

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Paulo Roberto Dellani

**DESENVOLVIMENTO DE UM SERVIDOR DE
IMAGENS MÉDICAS DIGITAIS NO PADRÃO
DICOM**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação

Professor Dr.rer.nat Aldo von Wangenheim


Florianópolis, fevereiro de 2001.

DESENVOLVIMENTO DE UM SERVIDOR DE IMAGENS MÉDICAS DIGITAIS NO PADRÃO DICOM

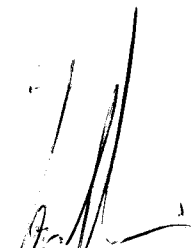
Paulo Roberto Dellani

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

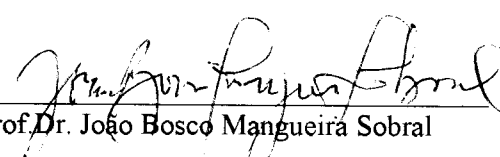
Banca Examinadora




Prof. Dr. Fernando A. O. Gauthier - Coordenador



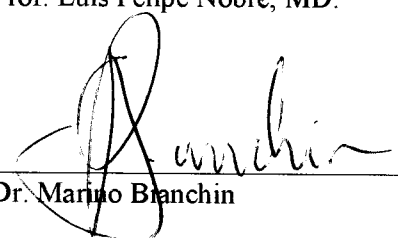
Prof. Dr. rer. nat. Aldo von Wangenheim - Orientador



Prof. Dr. João Bosco Manguiera Sobral



Prof. Luis Felipe Nobre, MD.



Dr. Marino Branchin

*“Quando o homem tiver conquistado todas as
profundidades do espaço e todos os mistérios
do tempo, então ele estará apenas
começando.”*

H.G.Wells, *The Shape of Things to come*

*A meus pais, Antonio Dellani e Lurdes
Maria Dellani, e familiares.*

AGRADECIMENTO

*Aos colegas e amigos, integrantes ou não, do
Projeto Cyclops, que de maneira ou outra
contribuíram para a realização deste.*

*A CAPES/ACAFE/PIQDT pelo auxílio
financeiro.*

Sumário

LISTA DE FIGURAS	X
LISTA DE TABELAS	XI
RESUMO	XII
ABSTRACT	XIII
1 - INTRODUÇÃO	1
1.1 - O PROJETO CYCLOPS	4
1.2 - MOTIVAÇÃO	4
1.3 - OBJETIVOS	5
1.3.1 - OBJETIVO GERAL	5
1.3.2 - OBJETIVOS ESPECÍFICOS	5
2 - CONCEITOS DE IMAGENOLOGIA MÉDICA DIGITAL	7
2.1 - ACR/NEMA 1.0 E ACR/NEMA 2.0	7
2.2 - DICOM 3	8
2.3 - PYPYRUS	11
2.4 - PACS	12
2.5 - TELEMEDICINA	12
2.6 - SERVIDORES DE IMAGENS MÉDICAS DIGITAIS	14
2.6.1 - CTN	14
2.6.2 - EFILM	16
2.6.3 - SOLUÇÕES PROPRIETÁRIAS: GE ADVANCED WINDOWS 3.1	17
2.7 - ANÁLISE DO ESTADO DA ARTE: POR QUE MAIS UM SERVIDOR?	17
3 - ENTENDENDO O PADRÃO DICOM 3	19
3.1 - PROCESSAMENTO DISTRIBUÍDO	19
3.2 - CONCEITOS GERAIS SOBRE DICOM	23
3.2.1 - CLASSES DE SERVIÇO E CLASSES SOP	24
3.2.2 - DEFINIÇÕES DE OBJETOS DE INFORMAÇÃO	25
3.2.3 - ATRIBUTOS	27

3.2.4 - ELEMENTOS DE SERVIÇO _____	28
3.2.4.1 - SERVIÇOS DE ASSOCIAÇÃO _____	28
3.2.4.2 - SERVIÇOS DIMSE _____	28
3.2.5 - INSTÂNCIAS SOP _____	30
3.2.6 - IDENTIFICAÇÃO _____	30
3.2.7 - RELAÇÕES _____	31
3.2.8 - REPRESENTAÇÃO DE VALOR _____	31
3.2.9 - SINTAXE DE TRANSFERÊNCIA _____	32
3.3 - CONCEITOS DE REDE DICOM _____	32
3.3.1 - ENTIDADE DE APLICAÇÃO _____	32
3.3.2 - ENDEREÇO DE APRESENTAÇÃO _____	34
3.3.3 - NEGOCIAÇÃO DE ASSOCIAÇÃO _____	34
3.3.3.1 – CONTEXTO DE APLICAÇÃO _____	34
3.3.3.2 - CONTEXTO DE APRESENTAÇÃO _____	35
3.3.3.3 – NOTIFICAÇÃO DE COMPRIMENTO MÁXIMO DE PDU _____	35
3.3.3.4 - NOTIFICAÇÃO DE IDENTIFICAÇÃO DE IMPLEMENTAÇÃO _____	37
3.4 - TIPOS DE CLASSE DE SERVIÇO DICOM _____	39
3.4.1 - VERIFICATION SERVICE CLASS _____	39
3.4.2 - STORE SERVICE CLASS _____	39
3.4.3 - MEDIA STORAGE SERVICE CLASS _____	40
3.4.4 - QUERY/RETRIEVE SERVICE CLASS _____	40
3.4.5 - STUDY CONTENT NOTIFICATION SERVICE CLASS _____	40
3.4.6 - PATIENT MANAGEMENT SERVICE CLASS _____	40
3.4.7 - STUDY MANAGEMENT SERVICE CLASS _____	41
3.4.8 - RESULTS MANAGEMENT SERVICE CLASS _____	41
3.4.9 - PRINT MANAGEMENT _____	41
4 - SUPORTE PARA COMUNICAÇÃO EM REDE DICOM _____	42
4.1 - AMBIENTE DE SUPORTE À COMUNICAÇÃO EM REDE _____	42
4.2 - SERVIÇOS DE CAMADA SUPERIOR OSI PARA ENTIDADES DE APLICAÇÃO _____	43
4.2.1 - SERVIÇO A-ASSOCIATE _____	44
4.2.1.1 - PROCEDIMENTOS DO SERVIÇO A-ASSOCIATE _____	44

4.2.2 - SERVIÇO A-RELEASE _____	46
4.2.2.1 – PROCEDIMENTOS DO SERVIÇO A-RELEASE _____	46
4.2.3 - SERVIÇO A-ABORT _____	48
4.2.3.1 - PROCEDIMENTOS DO SERVIÇO A-ABORT _____	48
4.2.4 - SERVIÇO A-P-ABORT _____	49
4.2.4.1 - PROCEDIMENTO DO SERVIÇO A-P-ABORT _____	50
4.2.5 - SERVIÇO P-DATA _____	50
4.3 - PROTOCOLO DE CAMADA SUPERIOR DICOM PARA TCP/IP _____	50
4.3.1 - USO DO SERVIÇO DE TRANSPORTE PROVIDO PELO TCP _____	51
4.3.1.1 - ABRINDO UMA CONEXÃO DE TRANSPORTE TCP _____	51
4.3.1.2 - TRANSFERINDO DADOS ATRAVÉS DE UMA CONEXÃO DE TRANSPORTE _____	52
4.3.1.3 - FECHANDO UMA CONEXÃO DE TRANSPORTE TCP _____	52
4.3.1.4 - CRONÔMETRO ARTIM _____	52
4.3.2 - PROTOCOLO DE CAMADA SUPERIOR PARA A MÁQUINA DE ESTADOS TCP/IP _____	53
4.3.3 - PROTOCOLO DA CAMADA SUPERIOR DICOM PARA ESTRUTURA DAS UNIDADES DE DADOS TCP/IP _____	58
5 - IMPLEMENTANDO O SERVIDOR DE IMAGENS MÉDICAS DIGITAIS	59
5.1 - CYCLOPS DICOM EDITOR _____	59
5.2 - ANÁLISE EM FUNÇÃO DOS OBJETIVOS _____	60
5.2.1 - PRIMEIRO OBJETIVO ESPECÍFICO _____	60
5.2.2 - SEGUNDO OBJETIVO ESPECÍFICO _____	62
5.2.3 - TERCEIRO OBJETIVO ESPECÍFICO _____	62
5.2.4 - QUARTO OBJETIVO ESPECÍFICO _____	62
5.3 - RESULTADOS OBTIDOS _____	63
5.3.1 - ESTABELECIMENTO DE ASSOCIAÇÕES COM OUTRAS DICOM AE__	63
5.3.2 - TRANSFERÊNCIA DE IMAGENS DE TOMOGRAFIA COMPUTADORIZADA _____	63
5.3.3 – VERIFICATION SERVICE CLASS _____	64
5.4 - DISCUSSÃO _____	64

6 - CONCLUSÕES	66
ANEXO 1 – CÓDIGO EM SMALLTALK PARA INICIAR O SERVIDOR	67
ANEXO 2 - SAÍDA DO COMANDO “STORESCU”	68
ANEXO 3 – SAÍDA DO COMANDO “SEND_IMAGE”	71
ANEXO 4 – SAÍDA DO COMANDO “ECHOSCU”	75
ANEXO 5 – SAÍDA DO COMANDO “DICOM_ECHO”	76
ANEXO 6 – CÓDIGO EM SMALLTALK PARA PARAR O SERVIDOR	79
7 - GLOSSÁRIO	80
8 - REFERÊNCIAS BIBLIOGRÁFICAS	82

Lista de Figuras

FIGURA 1: SIEMENS SIRETOM HEAD CT SCANNER_____	1
FIGURA 2: UM PACS COM SEUS PRINCIPAIS COMPONENTES_____	3
FIGURA 3: ARQUITETURA DE PROTOCOLOS DO DICOM 3 _____	8
FIGURA 4: MODELO DE PROCESSOS DISTRIBUÍDOS_____	19
FIGURA 5: CENÁRIO DE PROCESSOS DISTRIBUÍDOS _____	20
FIGURA 6: MODELO DE PROCESSOS DISTRIBUÍDOS_____	21
FIGURA 7: MODELO DE PROCESSOS DISTRIBUÍDOS DICOM_____	22
FIGURA 8: PRINCIPAIS ESTRUTURAS DO MODELO DE INFORMAÇÃO DICOM_____	23
FIGURA 9: DEFINIÇÕES DE OBJETOS DE INFORMAÇÃO E SUAS RELAÇÕES _____	25
FIGURA 10: EXEMPLO DE UM IOD COMPOSTO_____	26
FIGURA 11: DICOM EM REDE_____	33
FIGURA 12: NEGOCIAÇÃO DE CONTEXTOS DE APRESENTAÇÃO _____	36
FIGURA 13: NOTIFICAÇÃO DE COMPRIMENTO MÁXIMO DE PDU _____	37
FIGURA 14: NOTIFICAÇÃO DE UID DE CLASSE DE IMPLEMENTAÇÃO ____	38
FIGURA 15: NOTIFICAÇÃO DE NOME DE VERSÃO DE IMPLEMENTAÇÃO _	38
FIGURA 16: ARQUITETURA DE PROTOCOLOS DO DICOM 3 _____	43
FIGURA 17: ASSOCIATE REQUEST _____	44
FIGURA 18: ASSOCIATION RELEASE _____	46
FIGURA 19: TERMINO ANORMAL DA ASSOCIAÇÃO INICIADO PELO USUÁRIO DO SERVIÇO _____	48
FIGURA 20: TERMINO ANORMAL DA ASSOCIAÇÃO INICIADO PELO PROVEDOR DO SERVIÇO _____	49
FIGURA 21: TRANSFERÊNCIA DE DADOS_____	50
FIGURA 22: TELA DO CYCLOPS DICOM EDITOR_____	60
FIGURA 23: TELA DO CYCLOPS DICOM SERIES EDITOR _____	61

Lista de Tabelas

TABELA 1: SERVIÇOS DIMSE_____	29
TABELA 2: SERVIÇOS DA CAMADA SUPERIOR _____	44
TABELA 3: NENHUMA ASSOCIAÇÃO_____	53
TABELA 4: ESTABELECIMENTO DE ASSOCIAÇÃO_____	53
TABELA 5: TRANSFERÊNCIA DE DADOS _____	53
TABELA 6: LIBERAÇÃO DE ASSOCIAÇÃO _____	53
TABELA 7: FECHAMENTO DA CONEXÃO DE TRANSPORTE _____	54
TABELA 8: AÇÕES RELACIONADAS AO ESTABELECIMENTO DE UMA ASSOCIAÇÃO _____	54
TABELA 9: AÇÕES RELACIONADAS À TRANSFERÊNCIA DE DADOS _____	55
TABELA 10: AÇÕES RELACIONADAS À LIBERAÇÃO DE ASSOCIAÇÃO ____	55
TABELA 11: AÇÕES RELACIONADAS À TÉRMINO ANORMAL (ABORT) DE ASSOCIAÇÃO _____	56
TABELA 12: TRANSIÇÃO DE ESTADOS DO PROTOCOLO DE CAMADA SUPERIOR DICOM_____	57

Resumo

Em meados dos anos 70, com a Tomografia Computadorizada, surgiram as modalidades de exame médico que produziam imagens em formato digital. A utilização destas modalidades de exame médico digital em hospitais e clínicas radiológicas, fez necessário o desenvolvimento de sistemas de armazenamento e transmissão dos resultados produzidos em cada modalidade de exame - PACS (do inglês Picture Archiving and Communications Systems). Cada fabricante desenvolveu uma solução proprietária para a interconexão em PACS de seus próprios aparelhos de exames médicos digitais. Os diferentes padrões de codificação e transmissão dos resultados dos exames (imagens, dados do paciente, medições, etc.) adotados por cada fabricante, impossibilitavam a interconexão direta de aparelhos de diferentes fabricantes. Para resolver este problema, o NEMA e o ACR propuseram, no início da década de 80, o desenvolvimento de um padrão para PACS. Como resultado dessa iniciativa, atualmente temos o padrão DICOM (acrônimo do inglês Digital Imaging and Communications in Medicine). O objeto desta dissertação é o desenvolvimento e implementação de um software servidor, de acordo com as técnicas de codificação e transmissão definidas no padrão DICOM, para o armazenamento de imagens médicas digitais de diferentes modalidades de exames médicos.

Abstract

The medical screening modalities that produce digital images have appeared in the middle of the seventies, with the Computed Tomography. With the use of these digital medical screening modalities in radiological clinics and hospitals, the development of a Picture Archiving and Communications Systems (PACS) become necessary. Each manufacturer has developed proprietary PACS solutions to interconnect their own digital medical screening devices. The various screening results (images, patient data, measurements, etc.) coding and transmission proprietary techniques developed by every manufacturer haven't permitted the direct interconnection of medical screening devices from different manufacturers. To overcome these PACS interconnection problems, NEMA and ACR proposed the development of a PACS standard in the early eighties. As result, presently we have the DICOM - Digital Imaging and Communications in Medicine - PACS standard. The main goal of this work is the development of a server software, in accordance with the coding and transmission techniques specified by the DICOM standard, to store digital medical images from different medical screening modalities.

1 - Introdução

Durante a década de 70, uma nova modalidade de aparelhos para diagnóstico médico, a Tomografia Computadorizada (figura 1), introduziu o uso de técnicas de arquivamento e transmissão de imagens digitais nos ambientes clínico-hospitalares. Após a Tomografia Computadorizada outras modalidades de diagnóstico por imagens digitais surgiram, e o crescente uso de computadores em aplicações clínicas fez com que diversos padrões de codificação, arquivamento e transmissão de imagens e informações correlatas surgissem. A inexistência de um padrão comum trazia uma série de problemas ao interconectar sistemas de diferentes fornecedores. Cada fornecedor criava sua própria solução para o arquivamento, impressão, transmissão, visualização, etc. de imagens digitais e informações relacionadas de cada modalidade de exame.

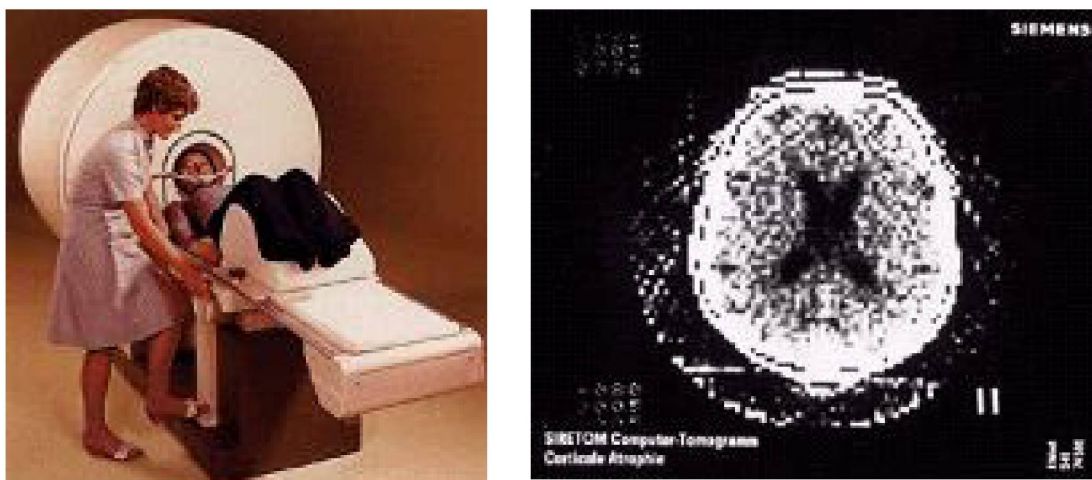


FIGURA 1: SIEMENS Siretom Head CT Scanner

Ter em uma clínica ou hospital com aparelhos de várias modalidades de exames e de diferentes fornecedores, um sistema ao qual todos estivessem conectados e pudessem uns compartilhar de recursos disponíveis aos outros, como uma impressora ou uma estação de visualização de imagens, era algo utópico. Sistemas de Arquivamento e Comunicação de Imagens (PACS, do inglês Picture Archiving and Communications Systems) eram soluções proprietárias, nas quais o acesso a informações sobre os

padrões e protocolos utilizados para codificação, armazenamento, transmissão, etc. das imagens e dados relacionados era restrito aos seus fornecedores/fabricantes.

O ACR (do inglês American College of Radiology - Colégio Americano de Radiologia) e o NEMA (do inglês National Electrical Manufacturers Association - Associação de Fabricantes de Equipamentos Elétricos Nacional dos EUA) reconheceram a necessidade de um método padrão para a transferência de imagens e informações correlatas entre dispositivos fabricados por diferentes fornecedores. O ACR e o NEMA formaram um comitê conjunto em 1983 a fim de desenvolver um padrão para:

- Promover a comunicação de informações de imagens digitais, independentemente do fabricante do aparelho;
- Facilitar o desenvolvimento e expansão dos PACS, como também a sua conexão a outros HIS (do inglês Hospital Information Systems - Sistemas de Informações Hospitalares);
- Permitir a criação de bases de dados de informações diagnósticas que podem ser consultadas por uma larga variedade de aparelhos geograficamente distribuídos.

A ACR-NEMA Standards Publication No. 300-1985, publicada em 1985 foi chamada de versão 1.0. Posteriormente o padrão foi revisado ainda duas vezes: em outubro de 1986 e em janeiro de 1988. A ACR-NEMA Standards Publication No. 300-1988 foi chamada de versão 2.0. Essas publicações padrão forneciam especificações para uma interface de hardware, um conjunto mínimo de comandos de software e um conjunto consistente de formatos de dados. Esse padrão sofria de várias deficiências, principalmente de não ter sido desenvolvido para suportar os ambientes de rede de computadores já disponíveis nas clínicas e hospitais na época de sua publicação.

Esse padrão permaneceu até os idos de 1992-1993, quando foi publicada uma nova versão radicalmente revisada mas mantendo compatibilidade com a anterior, sob o nome de ACR/NEMA Standards Publication PS3, também referenciada como DICOM 3 (do inglês Digital Imaging and Communications in Medicine). Após três anos de trabalho, com muitas sugestões da indústria e da área acadêmica, foi dado por completo

o padrão DICOM 3.0. Esta versão do padrão é muito mais abrangente e robusta que as versões anteriores, praticamente um novo projeto.

Atualmente o DICOM 3 é o padrão de fato para PACS, sendo suportado pela grande maioria dos aparelhos que trabalham com informações médicas digitais. O fato de um aparelho o suportar é a garantia de que este poderá ser facilmente integrado em um PACS já existente, devido à utilização de tecnologias de rede acessíveis e baratas (TCP/IP ou OSI) para a sua implementação, e ao fato deste poder utilizar-se de serviços disponibilizados por outros aparelhos que também suportem o padrão DICOM. Sua abrangência em PACS vai desde a codificação dos dados das imagens e informações associadas, passando pela definição de diversas classes de serviços, como armazenamento, recuperação, pesquisa e impressão de imagens, formatos utilizados no armazenamento das imagens em meios removíveis, processos de negociação de associações para a transmissão dos dados das imagens através de redes, etc. Softwares aplicativos para a área médica também podem obter vantagens ao oferecer suporte ao padrão, pelas mesmas razões acima citadas. Na figura 2 podemos ver um exemplo de um PACS com seus componentes principais.

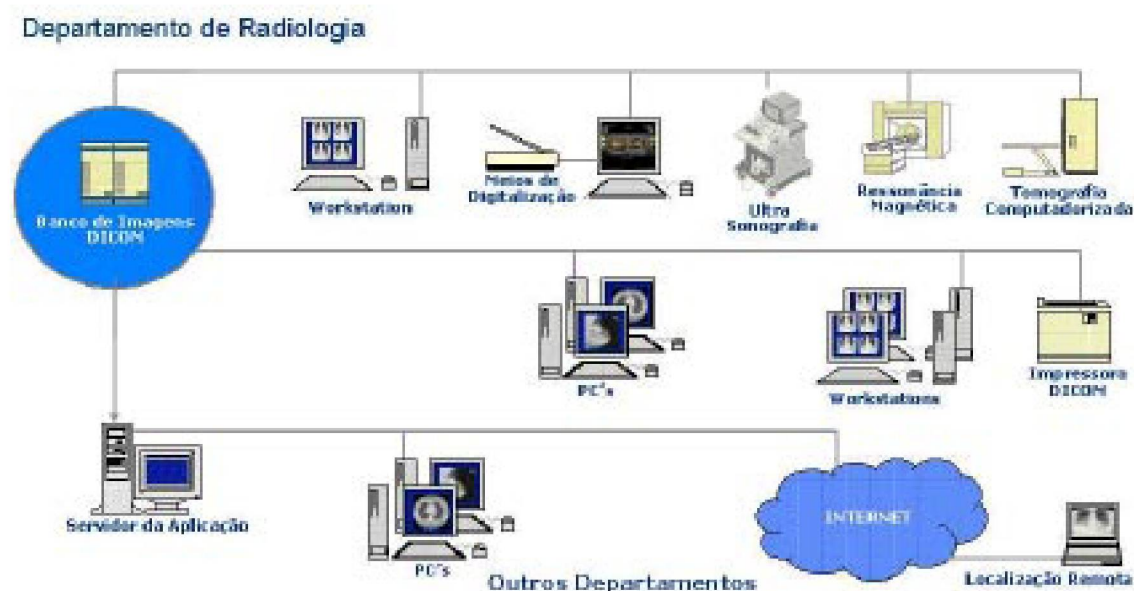


FIGURA 2: Um PACS com seus principais componentes

1.1 - O Projeto Cyclops

O projeto Cyclops é um projeto binacional de pesquisa de longo prazo iniciado pelos Professores Dr.rer.nat.Aldo von Wangenheim e Dr.Michael M.Richter na Universidade de Kaiserslautern - Alemanha, em 1992. Tem como objetivo o desenvolvimento e a transferência de novos métodos, técnicas e ferramentas no campo da Análise de Imagens Médicas através da utilização de técnicas de Inteligência Artificial e Visão Computacional.

Neste contexto, a cooperação com parceiros médicos e industriais foi iniciada em 1993. Hoje o Projeto se encontra em sua Fase II, estando focado na cooperação para o desenvolvimento de aplicações que possam ser de utilidade prática clínica, dentro de um consórcio de nível internacional.

Este consórcio pretende alcançar as metas do Projeto através da cooperação entre os parceiros do Brasil e da Alemanha, cujas competências específicas complementam-se em áreas relevantes ao mesmo. O referido consórcio internacional de pesquisa e desenvolvimento é composto por Universidades, parceiros industriais da área de “software”, parceiros médicos e empresas produtoras de equipamentos médicos radiológicos de ambos os países.

1.2 - Motivação

A presente dissertação de mestrado é fruto da necessidade por parte dos softwares aplicativos para a área médica desenvolvidos no contexto do Projeto Cyclops terem acesso às informações médicas digitais existentes em PACS. Esses aplicativos devem para tanto ter suporte ao padrão DICOM.

Atualmente os aplicativos do Projeto Cyclops utilizam um software chamado de “DICOM Editor” para ler imagens médicas digitais e informações correlatas que estejam armazenadas em determinados servidores (falando em DICOM, Entidades de Aplicação para Armazenamento de Imagens Médicas Digitais e outras Informações Médicas). Portanto, a instalação destes aplicativos requer a existência de um servidor de imagens médicas digitais já em funcionamento (ou a instalação de um novo), no respectivo ambiente de utilização. Com o software - cujo desenvolvimento é descrito

nesta dissertação – será possível o envio de imagens médicas digitais e outras informações diretamente para a aplicação que as processará, sem a necessidade de armazenamento em outros dispositivos.

Sistemas para a conversão automática de séries inteiras de imagens de pacientes, no formato DICOM 3, para outros formatos digitais de arquivos de imagem mais compactos e facilmente transmissíveis através de linhas de dados de baixa velocidade, conectadas à Internet poderiam ser implementados com bastante facilidade, tendo em mãos um software capaz de receber imagens diretamente de um aparelho de Ultra Som ou Tomografia Computadorizada, por exemplo, sem a dependência de soluções de terceiros, que muitas vezes tem limitações ou características de desenvolvimento que dificultam muito a sua expansão.

Um Servidor como este facilitaria em muito a implantação e aumentaria a receptividade dos parceiros médicos aos softwares desenvolvidos dentro do projeto na área de Informática Médica.

1.3 - Objetivos

Os objetivos deste trabalho foram divididos em Objetivo Geral e Objetivos Específicos, a fim de facilitar a análise do problema a ser resolvido e a implementação.

1.3.1 - Objetivo geral

Implementar um protótipo de servidor para PACS que atenda às classes de serviço de armazenamento de imagens e outras informações relacionadas de acordo com os métodos definidos pelo padrão DICOM 3, em ambientes de rede TCP/IP, compatível com os aplicativos já desenvolvidos dentro do Projeto Cyclops.

1.3.2 - Objetivos específicos

Para que se possa implementar um protótipo de servidor para PACS que atenda às classes de serviço de armazenamento de imagens e outras informações relacionadas de acordo com os métodos definidos pelo padrão DICOM 3, os objetivos específicos abaixo enumerados devem ser atingidos.

- 1) Modelar e implementar a camada de serviços denominada pelo padrão de “DICOM Upper Layer protocol for TCP/IP”;
- 2) Modelar e implementar as funções comuns ao que o padrão denomina de “DICOM Application Entities”;
- 3) Implementar o suporte necessário ao que o padrão denomina de “DICOM Model of the Real World”;
- 4) Implementar a classe de serviços de armazenamento, denominada pelo padrão de “DICOM Image Storage Services”.

2 - Conceitos de imagenologia médica digital

O ramo de imagens médicas digitais tem uma série de termos técnicos e padrões, cujo esclarecimento prévio faz-se necessário antes da continuidade da leitura dos demais capítulos desta dissertação.

2.1 - ACR/NEMA 1.0 e ACR/NEMA 2.0

A iniciativa de padronização para PACS foi tomada pelo ACR (do inglês American College of Radiology) e pelo NEMA (do inglês National Electrical Manufacturers Association) no início dos anos 80 (1983), e em 1985 a versão 1.0 do padrão foi publicada, sendo então referenciada como ACR/ NEMA Standards Publication No. 300-1985. O padrão foi revisados várias vezes e em 1988 foi publicada a versão 2.0, descrita no ACR/NEMA Standards Publication No. 300-1988. Assim, em 1988, a codificação e intercâmbio de imagens médicas digitais começou a ser verdadeiramente padronizada. Apesar de começar a ser largamente utilizada, esta segunda versão do padrão ainda não fornecia uma comunicação confiável, em virtude das diversas modificações incorporadas à esta versão [CLUNIE, 2000].

Na verdade, o problema foi que em 1988 muitos usuários queriam uma interface entre seus dispositivos de imagem e suas redes. Apesar disto já ser permitido pela versão corrente do padrão, o mesmo ainda necessitava de melhoramentos a fim de permitir uma comunicação em rede robusta. Por exemplo, um dispositivo poderia enviar uma mensagem contendo a imagem e seu cabeçalho para um outro dispositivo qualquer, mas este último não tinha como saber o que o primeiro queria fazer com as informações contidas na mensagem. Isso acontecia pois o padrão de 1988 não foi desenvolvido exatamente para suportar comunicação em rede. Este ponto ainda era secundário para o padrão.

Este padrão deve sua importância principalmente ao fato de que muitos formatos proprietários e outros padrões *de facto* adotaram o formato de mensagens do ACR/NEMA e seu correspondente dicionário de dados e mecanismos de expansão. [CLUNIE, 2000].

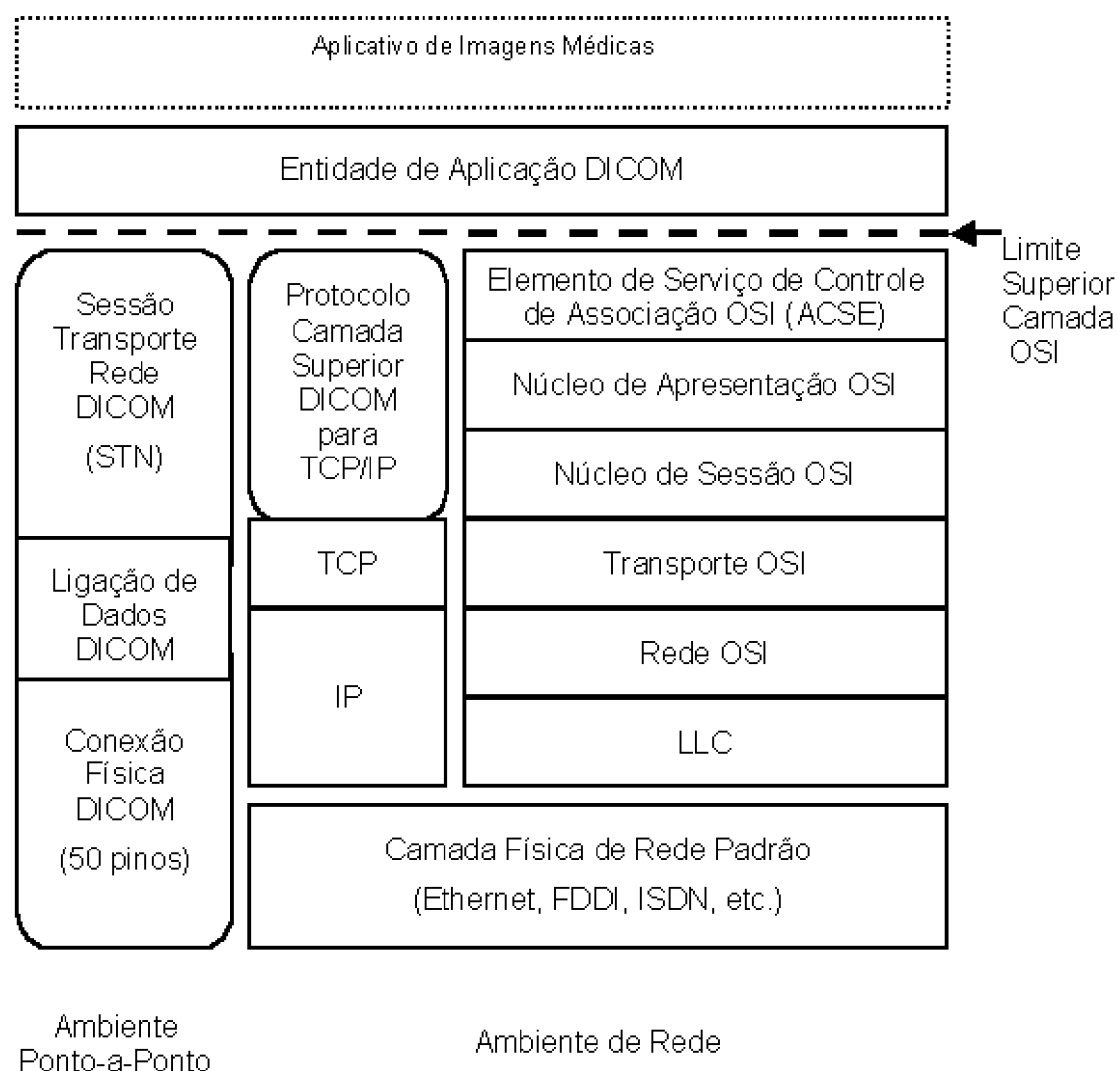


FIGURA 3: Arquitetura de protocolos do DICOM 3

2.2 - DICOM 3

Entre 1992-1993 foi publicada uma nova versão radicalmente revisada, mas compatível com o padrão anterior, sob o nome de ACR/NEMA Standards Publication PS3, também referenciada como DICOM 3 (acrônimo do inglês Digital Imaging and Communications in Medicine). Adicionalmente, um rápido exame dos tipos de serviços necessários à comunicação em rede, mostrou que a definição de uma classe de serviços básicos permitiria que um processo de alto nível (na camada de aplicação) fosse capaz de se comunicar com um grande número de diferentes protocolos de rede. Assim, deu-

se o desenvolvimento de uma nova versão, quase um novo projeto. Estes protocolos são modelados como uma série de camadas superpostas, também conhecidas como “pilha” [CLUNIE, 2000].

Na versão 2.0 do DICOM já existia uma “pilha” que definia uma comunicação ponto-a-ponto. Mais tarde foram inseridas, baseadas em sua popularidade e possibilidade de expansão, as pilhas de protocolos de rede TCP/IP (do inglês Transmission Control Protocol/Internet Protocol) e a ISO/OSI (do inglês International Standards Organization/Open Systems Interconnection).

A figura 3 apresenta um diagrama do modelo de comunicação desenvolvido. A filosofia básica é que uma dada aplicação de imagens médicas possa se comunicar sobre qualquer das pilhas disponíveis com qualquer outro dispositivo que utilize a mesma pilha. Com esta nova filosofia, tornou-se possível a troca de pilhas, sempre que necessário, sem ter de rescrever todo o código dos programas aplicativos que utilizassem o padrão DICOM. Após três anos de trabalho, com muitas sugestões da indústria e da área acadêmica, foi dado por completo o DICOM 3.0. Esta versão é muito mais abrangente e robusta que as versões anteriores.

Atualmente o DICOM 3 é o padrão *de facto* para PACS, sendo suportado pela grande maioria dos aparelhos de diagnóstico médico. O fato de um aparelho o suportar é a garantia de que este poderá ser facilmente integrado em um PACS já existente, devido a utilização de tecnologias de rede acessíveis e baratas (TCP/IP ou OSI) para a sua implementação. Sua abrangência em PACS vai desde a codificação dos dados das imagens, passando pela definição de diversas classes de serviços, como armazenamento, recuperação, pesquisa e impressão de imagens, formatos utilizados no armazenamento das imagens em meios removíveis, processos de negociação de associações para a transmissão dos dados das imagens através de redes, etc. A sua extensão pode ser vista na maneira como foram divididos os capítulos que definem o padrão, na sua última versão publicada (2000):

- **PS 3.1-2000** *Introduction and Overview*: uma breve introdução ao padrão DICOM.
- **PS 3.2-2000** *Conformance*: define os princípios para todas as implementações que busquem estar em conformidade com o padrão DICOM.

- **PS 3.3-2000** *Information Object Definitions*: descreve como os objetos de informação são construídos. Lista todos os grupos de Entidades de Informação e Definições de Objetos de Informações. Estes grupos são coleções e seqüências de elementos de dados.
- **PS 3.4-2000** *Service Class Specifications*: contém a especificação das Classes de Serviço. Explica como várias classes de serviço são implementadas pelas entidades de aplicação. As regras de SCU (usuário do serviço) e SCP (fornecedor do serviço) também são definidas nesta parte. É uma boa referência/tutorial para programadores de aplicações DICOM.
- **PS 3.5-2000** *Data Structures and Encoding*: especifica o método de codificação de dados para a transmissão e decodificação após a recepção dos dados.
- **PS 3.6-2000** *Data Dictionary*: provê uma lista com todas as Tags correspondentes aos atributos dos objetos de informação que fazem parte do padrão.
- **PS 3.7-2000** *Message Exchange*: descreve os Serviços DICOM comumente referenciados como DIMSE-C e DIMSE-N (do inglês DICOM Message Service Element - Composite / Normalized, respectivamente). Estes serviços são encapsulados pela comunicação DICOM.
- **PS 3.8-2000** *Network Communication Support for Message Exchange*: especifica o processo de negociação de associação, protocolos da camada superior, transporte dos dados e estado da máquina. **Este é o componente mais complexo para implementadores DICOM.**
- **PS 3.9-2000** *Point to Point Communication Support for Message Exchange*: especifica a utilização do padrão DICOM sobre conexões ponto-a-ponto.
- **PS 3.10-2000** *Media Storage and File Format for Media Exchange*;
- **PS 3.11-2000** *Media Storage Application Profiles*;
- **PS 3.12-2000** *Media Formats and Physical Media for Media Interchange*: explicam os formatos utilizados para armazenar dados em meios removíveis

como discos WORM, visando facilitar a transferência de informações entre sistemas computadorizados para imagens digitais em ambientes médicos não interligados em rede.

- **PS 3.13-2000** *Print Management Point-to-Point Communication Support*: especifica os serviços e protocolo necessários para suportar a comunicação de Entidades de Aplicação de Gerenciamento de Impressão DICOM.
- **PS 3.14-2000** *Grayscale Standard Display Function*: especifica uma função de exibição padronizada para a exibição de imagens em escala de cinza. Provê exemplos de métodos para mensuração da curva característica de um sistema de exibição em particular, com propósitos de calibração do mesmo, ou para a mensuração da conformidade de um sistema de exibição à função de exibição em escala de cinza padrão.
- **PS 3.15-2000** *Security Profiles*: aborda a conformidade de implementações utilizando o padrão DICOM quanto às questões de segurança.

2.3 - PAPYRUS

PAPYRUS é um formato de arquivo para o intercâmbio e armazenamento de imagens médicas e informações correlatas, baseado no padrão definido pelo ACR/NEMA. Foi desenvolvido para ser utilizado pelo OSIRIS, um software portátil e expansível para a manipulação iterativa de imagens médicas digitais. O software OSIRIS foi desenvolvido na Universidade Hospital de Genebra, com implementações para diferentes plataformas de hardware. O PACS implementado neste hospital utiliza-se principalmente do software OSIRIS.

A primeira versão do formato de arquivo PAPYRUS foi criada em 1990, baseada no dicionário de dados e na estrutura de dados do padrão de comunicação ACR/NEMA 2.0. Esse formato de arquivo foi uma resposta às necessidades de um formato para o armazenamento e a comunicação de imagens. Posteriormente, o PAPYRUS foi estendido e adotado por vários projetos europeus nas áreas de imagens digitais e teleradiologia. A Universidade/Hospital de Genebra fornece o código-fonte completo de uma biblioteca chamada de “PAPYRUS Toolkit” para facilmente ler e

escrever arquivos no formato PAPYRUS. O componente de formato de arquivo médico da versão 2 do PAPYRUS estendia o formato ACR/NEMA, particularmente no que diz respeito a referenciar múltiplas imagens ao colocar informações de pastas referenciando conjuntos de dados ACR/NEMA em um grupo oculto (privado) [OSIRIS, On-Line].

Em 1992 o comitê ACR/NEMA publicou versões prévias do completamente remodelado padrão DICOM. Os desenvolvedores do formato PAPYRUS estavam em contato com os novos desenvolvimentos do padrão DICOM, e atualizaram o formato de arquivos PAPYRUS, para que este correspondesse à nova estrutura de dados DICOM. Contribuindo com o desenvolvimento do padrão DICOM 3, o time de desenvolvimento do PAPYRUS 3 o está atualizando nesta versão, para ser compatível com o formato de arquivo previsto na parte 10 do padrão DICOM 3.

2.4 - PACS

PACS (do inglês Picture Archiving and Communications System) são sistemas compostos por aparelhos de diagnóstico médico que fornecem imagens digitais, como por exemplo CT (Tomografia Computadorizada), US (Ultra-Som) e MR (Ressonância Magnética), estações de trabalho para processamento das imagens, servidores de armazenamento de imagens médicas digitais e informações sobre pacientes, scanners, impressoras, filmadoras, etc., todos interconectados através de uma rede local/remota de dados. A finalidade de um sistema como este é possibilitar, através de softwares clientes rodando nas estações de trabalho locais e remotas visualizar, trabalhar, imprimir e efetuar outras operações com as imagens armazenadas nos servidores de imagens médicas digitais.

2.5 - Telemedicina

Conceitualmente, a telemedicina pode ser compreendida como a forma de se prestar assistência médica, utilizando-se a transmissão de imagens estáticas, vídeo, e várias formas de informações através dos meios de telecomunicação tais como: cabos, fibras óticas, satélites, Internet, etc.

As primeiras experiências no campo da transmissão de imagens em vídeo e som com finalidades médicas datam das missões espaciais soviéticas e americanas nos anos 60. Nos dias atuais, a Telemedicina é encarada como uma forma de difundir cuidados na área da saúde para localidades desprovidas dos mesmos, ou ainda, deficitárias de determinados tipos de procedimentos, com o objetivo amplo de permitir igualdade de acesso aos serviços médicos, independentemente da localização geográfica do indivíduo. Além desta importante atividade assistencial, o desenvolvimento da Telemedicina, em função do seu caráter marcadamente interativo, possibilita a atuação nas áreas de ensino e pesquisa, servindo-se de pontos estruturais inseridos no que poderíamos denominar como Universidade Virtual da Saúde.

Praticamente, quase todas as especialidades médicas podem vir a utilizar-se da telemedicina. É claro que algumas, particularmente aquelas que utilizam imagens como meio de diagnóstico, são as mais prontamente beneficiadas. Assim, os setores de radiologia, dermatologia, patologia, ultra-sonografia, entre outros, são bastante propícios para o estabelecimento de protocolos de transmissão de dados à distância, com finalidades diagnósticas. É Interessante notar que apesar da vocação natural destas disciplinas da área de imagenologia para a Telemedicina, a psiquiatria foi um dos primeiros campos a desenvolver aplicações nesse setor.

A utilização da infra-estrutura de comunicação previamente existente pode ser uma alternativa de custo-efetividade para a Telemedicina. As linhas telefônicas, desta forma, são os meios de conexão mais comuns entre pontos distantes. Outras possibilidades incluem a utilização de linhas especiais (ISDN / T1), modems de alta velocidade (DSL), fibras óticas ou mesmo conexões via satélite. As linhas ISDN (do inglês Integrated Services Digital Networks) permitem transmissão de dados a velocidades de 64 a 128 kbps, além de duas linhas de voz, podendo ser agregadas para aumentar-se a largura da banda de transmissão em múltiplos destes valores. As fibras óticas permitem transmissão de dados em velocidades superiores a 150 Mbps (megabits por segundo), sem sofrerem as interferências habituais relacionadas às linhas de transmissão do tipo metálico. Já o uso da comunicação via satélite envolve equipamentos de transmissão e recepção específicos, bem como a disponibilidade do canal no próprio satélite.

2.6 - Servidores de imagens médicas digitais

Um componente essencial de um PACS é o servidor de imagens médicas. É nele que as imagens médicas digitais e informações correlacionadas produzidas nos diversos aparelhos de modalidades de diagnóstico médico de um PACS são armazenadas. Existem várias soluções para servidores de imagens médicas digitais, desde as incluídas em pacotes de código aberto gratuitas até soluções comerciais e/ou proprietárias.

O ponto mais relevante a ser considerado em um servidor de imagens médicas digitais, é a sua capacidade de comunicação com aparelhos de diagnóstico médico de diferentes fornecedores, o que é comum no ambiente de uma clínica ou de um hospital. Os primeiros sistemas deste gênero eram soluções proprietárias e utilizavam-se de diferentes formatos de imagens médicas digitais e diferentes protocolos para a transmissão/arquivamento destas. Um servidor de imagens médicas digitais, em um sistema PACS envolvendo aparelhos de vários fabricantes, que fosse inteiramente funcional era algo bastante difícil de se implementar, pois era necessária a conversão dos diferentes formatos de imagens e a compatibilização dos protocolos de transmissão e arquivamento das mesmas. Essa tarefa beira a impossibilidade, pois a maioria dos fabricantes de aparelhos médicos não revelam detalhes do funcionamento interno destes sistemas, como formatos de arquivos, estrutura de gravação dos mesmos em mídias e os protocolos de transmissão destes através de redes de dados.

A facilidade de se implementar extensões a estes softwares também deve ser levada em conta no contexto de um projeto de pesquisa como o Cyclops. Como já foi dito anteriormente, várias novas aplicações médicas estão em desenvolvimento, com muitas funcionalidades cuja implementação baseada em pacotes de código aberto de implementação confusa, em linguagens não orientadas a objeto, é difícil, ou mesmo impossível, no caso de pacotes de software cujo código-fonte não o acompanha.

2.6.1 - CTN

O CTN (do inglês Central Test Node) é um pacote de software fornecido pelo ACR com a finalidade de realizar testes de conformidade em softwares que

implementem suporte a funcionalidades definidas no padrão DICOM. Outra finalidade do CTN é a realização de demonstrações envolvendo estas funcionalidades.

Para fazer cumprir com as finalidades para que foi desenvolvido, estão incluídas no CTN uma série de ferramentas de software capazes de atuar como, por exemplo, visualizadores de arquivos de imagens médicas digitais, servidores de imagens médicas digitais, gateways para HIS (do inglês Hospital Information Systems), servidores de impressão, emuladores de modalidades, utilitários para o envio e recebimento de imagens para/de entidades de aplicação, etc. É distribuído em forma de código-fonte, portanto tendo que ser especificamente configurado e compilado para a plataforma operacional na qual funcionará. Este processo é bastante complexo e envolve uma série de requisitos. As plataformas operacionais nas quais o CTN pode rodar são o Unix e o MS-Windows. Nesta última, o CTN não dispõe de algumas das funcionalidades.

Atualmente, no Projeto Cyclops, o CTN é utilizado como solução temporária para servidor de imagens médicas digitais, através do componente de software `image_server` (nome de um dos programas que acompanha o pacote). Este software tem uma série de limitações do ponto de vista de banco de dados, como por exemplo a inexistência de ferramentas para a manipulação dos dados armazenados no arquivo (manutenção da base de dados, cópias de segurança, etc.), inexistência de opções de gerenciamento do servidor, como por exemplo, o encerramento de operação do mesmo (shutdown), a verificação do estado dos registros de dados, etc. A experiência adquirida com a utilização deste software no projeto tem revelado que este também tem sérias limitações de performance, demorando muito mais do que softwares equivalentes, para efetuar operações de transferência de imagens.

Com todas as limitações desta ferramenta de software, a sua utilização em ambientes hospitalares ou de clínicas não pode ser realizada de uma maneira profissional, a princípio, pois essas deficiências deverão ser compensadas de alguma maneira, como por exemplo com pessoal extra treinado para efetuar manualmente as tarefas que deveriam estar implementadas de maneira automática na ferramenta.

2.6.2 - eFilm

Este software foi um dos melhores que surgiram até o momento para a plataforma de computadores PC, no que diz respeito ao padrão DICOM. Trata-se de um software que faz o papel de uma “estação de trabalho” para imagens médicas digitais no padrão DICOM. Algumas de suas funcionalidades:

- *Estação de Visualização*
- *Armazenamento e Transmissão*
- *Impressão DICOM*
- *Armazenamento em Mídias*

Como estação de visualização de imagens médicas digitais, o eFilm possui uma série de características notáveis, tais como a possibilidade de visualização simultânea de estudos de pacientes e suas respectivas imagens, fácil controle da “janela” utilizada na visualização da imagem com o mouse, conversão das imagens para outros formatos digitais de imagens (JPEG, TIFF, etc.), impressão das imagens em impressoras comuns, manipulação das informações sobre pacientes, estudos e séries de cada imagem no formato DICOM, transferência destas imagens e informações para outras entidades de aplicação DICOM via rede de computadores, etc.

O eFilm vem com um servidor DICOM com algumas funções de gerenciamento, tal com shutdown, reinicialização, definição de bandas de utilização do espaço em disco no sistema, monitoramento de filas de transferência de imagens de/para outras entidades de aplicação, etc. Conta também com algumas facilidades para a criação de CD-R e outras mídias de alta capacidade de armazenamento, com imagens selecionadas do seu arquivo. Pode-se incluir as imagens em formato DICOM, HTML+JPEG ou ambos em uma estrutura de diretórios para, posteriormente, serem transferidas para mídia, juntamente com o eFilm Lite, uma versão mais simples do eFilm.

Até o momento em que esta dissertação foi escrita, o mesmo ainda continuava a ser distribuído gratuitamente pela Internet, apesar de a partir da versão 1.5.0 esta distribuição ser feita através de uma empresa chamada de eFilm Inc., e não mais pela “University Health Network”.

2.6.3 - Soluções proprietárias: GE Advanced Windows 3.1

A estação de trabalho radiológica GE Advanced Windows 3.1 vem com um servidor de imagens médicas digitais DICOM, porém as únicas aplicações que conseguem armazenar imagens neste servidor são fabricados pela GE também. É possível o envio de imagens médicas digitais no formato DICOM a partir deste equipamento para outras Entidades de Aplicação DICOM em um PACS, mas não é possível enviar imagens médicas, por exemplo, do eFilm ou de um aparelho de ultrassom de outro fabricante, para o servidor embutido nesta estação de trabalho.

2.7 - Análise do estado da arte: por que mais um servidor?

Existem mundo afora várias opções de software para servidores de imagens médicas digitais no formato DICOM, cada qual com suas vantagens e desvantagens. O principal empecilho na utilização em larga escala destes no Brasil é o custo destas soluções. Sistemas comerciais/proprietários estão disponíveis apenas por preços que na maioria das vezes são difíceis de justificar, predominando em nossas clínicas e hospitais sistemas totalmente manuais para a análise diagnóstica de imagens produzidas por aparelhos de imagenologia médica capazes de trabalhar com o padrão DICOM 3.0.

Sistemas proprietários para o arquivamento de imagens médicas digitais, já embutidos em aparelhos de diagnóstico médico, ou vendidos em separado (e consequentemente caros e inacessíveis para a grande maioria das clínicas e hospitais brasileiros), não são totalmente compatíveis com o padrão DICOM 3.0, deixando de implementar muitas coisas previstas no padrão, não servindo ainda para implementações flexíveis de HIS e PACS.

Pacotes de código aberto como o CTN tem sérias limitações de performance, deixando a desejar quanto a certas funcionalidades de sistema para arquivamento de imagens médicas digitais, como a inexistência de ferramentas para administração da base de dados (cópias de segurança, verificação de integridade dos dados, etc.). Uma outra desvantagem do CTN é a sua implementação intrincada e em linguagem de programação não orientada a objetos, que dificulta a extensão de suas funcionalidades e tarefas de reengenharia de software sobre o mesmo. Requisitos para HIS são satisfeitos

muito precariamente no CTN também. No Projeto Cyclops muitas coisas novas estão sendo desenvolvidas e testadas, como por exemplo reconstruções em 3D, onde novos IODs (do inglês Information Object Definition – Definição de Objeto de Informação) devem ser suportados (VR IOD) [BRAMORSKI], e como já foi mencionado, em um software como o CTN essas extensões são muito complicadas de se implementar.

Softwares como o eFilm, distribuídos gratuitamente pela Internet, podem até ser utilizados, mas não possuem código-fonte aberto e estão sujeitos à vontade de seus desenvolvedores apenas, que de uma hora para outra podem começar a cobrar pela sua utilização, e demorar (e cobrar) para implementar funcionalidades requeridas por possíveis clientes “mais exigentes”, como seria o caso de um projeto como o Cyclops. Além disso, sua implementação atual é recheada de bugs de software, o que leva a acreditar que o principal motivo para a sua distribuição gratuita pela rede, até o momento, é a facilidade proporcionada por tal atitude no que tange a arrumar pessoas para efetuar testes do mesmo.

3 - Entendendo o Padrão DICOM 3

DICOM é um padrão que engloba diversos aspectos de imagenologia médica digital, desde a codificação dos dados que compõem uma imagem, parâmetros de visualização destas imagens, formato de arquivamento em disco, serviços para a comunicação de imagens e informações através de redes de computadores, etc. Neste capítulo serão abordados os diferentes aspectos que compõem o padrão DICOM.

3.1 - Processamento distribuído

Um modelo simples de processamento distribuído explica os mecanismos e a terminologia utilizada no padrão DICOM.

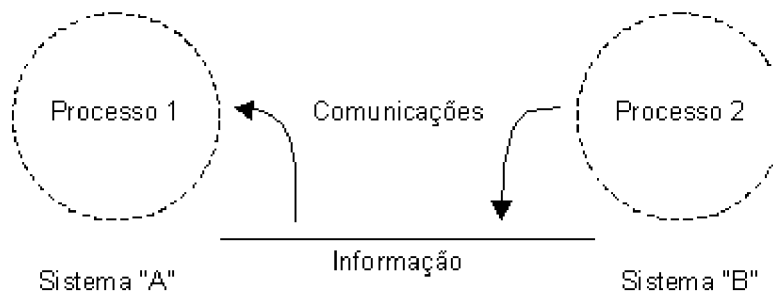


FIGURA 4: Modelo de processos distribuídos

Um *processo distribuído* tem pelo menos dois processos compartilhando informação, cada um fazendo seu processamento mas contando com as funcionalidades do outro lado. Um número de processos atuando em conjunto provêm serviços para sistemas em ambientes tais como departamentos de radiologia. Por exemplo, modalidades, servidores e estações de trabalho provêm serviços tais como aquisição, armazenamento e visualização de dados de imagens [REVET, 1997].

Na maioria dos cenários de processos distribuídos, os processos de aplicação são fortemente separados dos processos de comunicação que coordenam a transmissão de dados entre sistemas e compensam as diferentes maneiras nas quais os valores são representados internamente em diferentes sistemas.

Antes de os processos poderem atuar em conjunto, determinadas questões devem ser resolvidas. Eles tem de acordar sobre qual papel cada um irá desenvolver, ter uma visão equivalente da informação e selecionar as operações que cada lado implementa. A figura 6 mostra os detalhes [RENET, 1997].

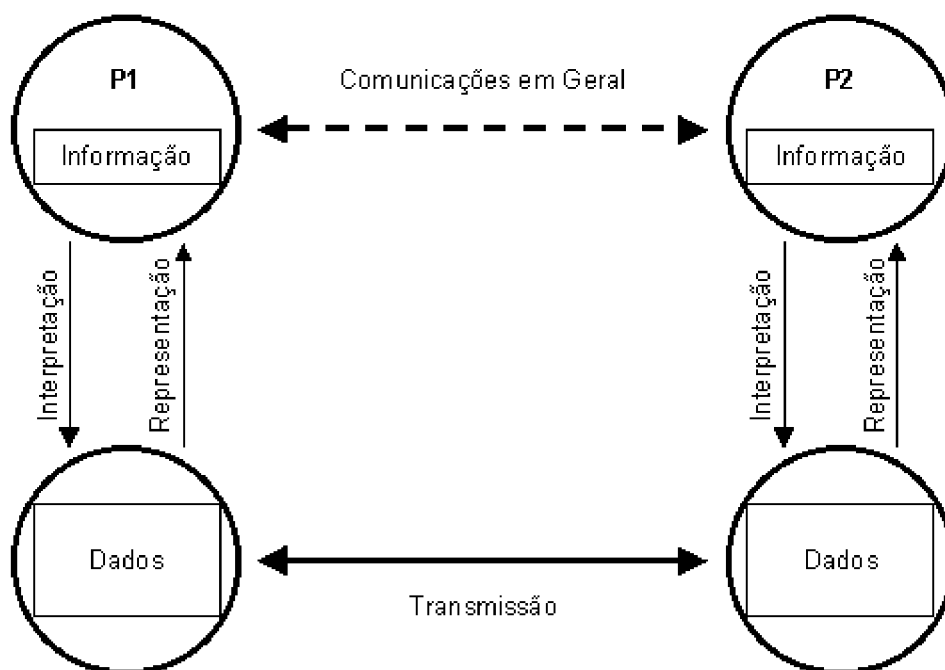


FIGURA 5: Cenário de processos distribuídos

Primeiramente o papel de cada lado tem de ser definido como cliente ou como servidor. O lado que usa as funcionalidades do outro lado tem o papel de cliente. O lado oposto atuando em um modelo concordado tem o papel de servidor. As expectativas que cada um tem do outro estão definidas no relacionamento que compartilham. O relacionamento define qual lado, e sob quais condições, toma a iniciativa no processo - na maioria das vezes é tomada pelo lado cliente.

Além disso, ambos os lados devem acordar sobre a informação que eles transferem. Aqui, a semântica da informação é considerada, não a maneira como ela é representada (sintaxe). A informação é definida pelo contexto do serviço que o processo distribuído está implementando. A operação define como a informação transferida deve ser processada no outro lado, tal como armazenar a informação, retornar um resultado, etc.

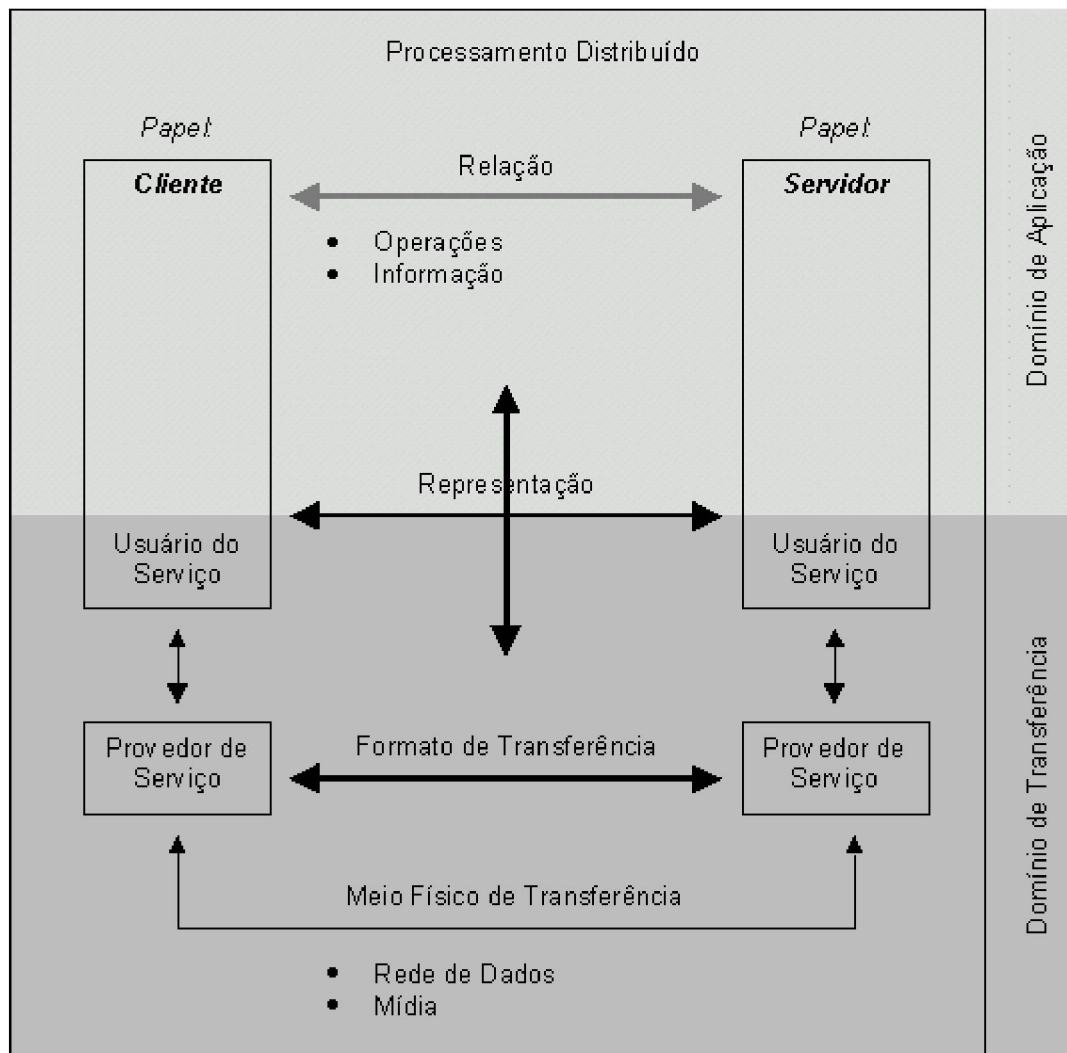


FIGURA 6: Modelo de processos distribuídos

A combinação de contexto, relacionamento, operações e informação é o ponto central do processamento distribuído e tem de ser determinada antes que uma implementação de sucesso possa ser realizada. Todas essas questões fazem parte do domínio de aplicação dos processos distribuídos (figura 6). Os processos não conhecem a maneira pela qual a informação é transferida, mas contam com serviços de nível mais baixo (por exemplo TCP/IP) providos no domínio de transferência para isso. Serviços de baixo nível manipulam a transferência de informações e estão escondidos do domínio de aplicação do cliente ou do servidor. A parte requerendo serviços é chamada de usuário do serviço. A contraparte é o provedor de serviço [REVET, 1997].

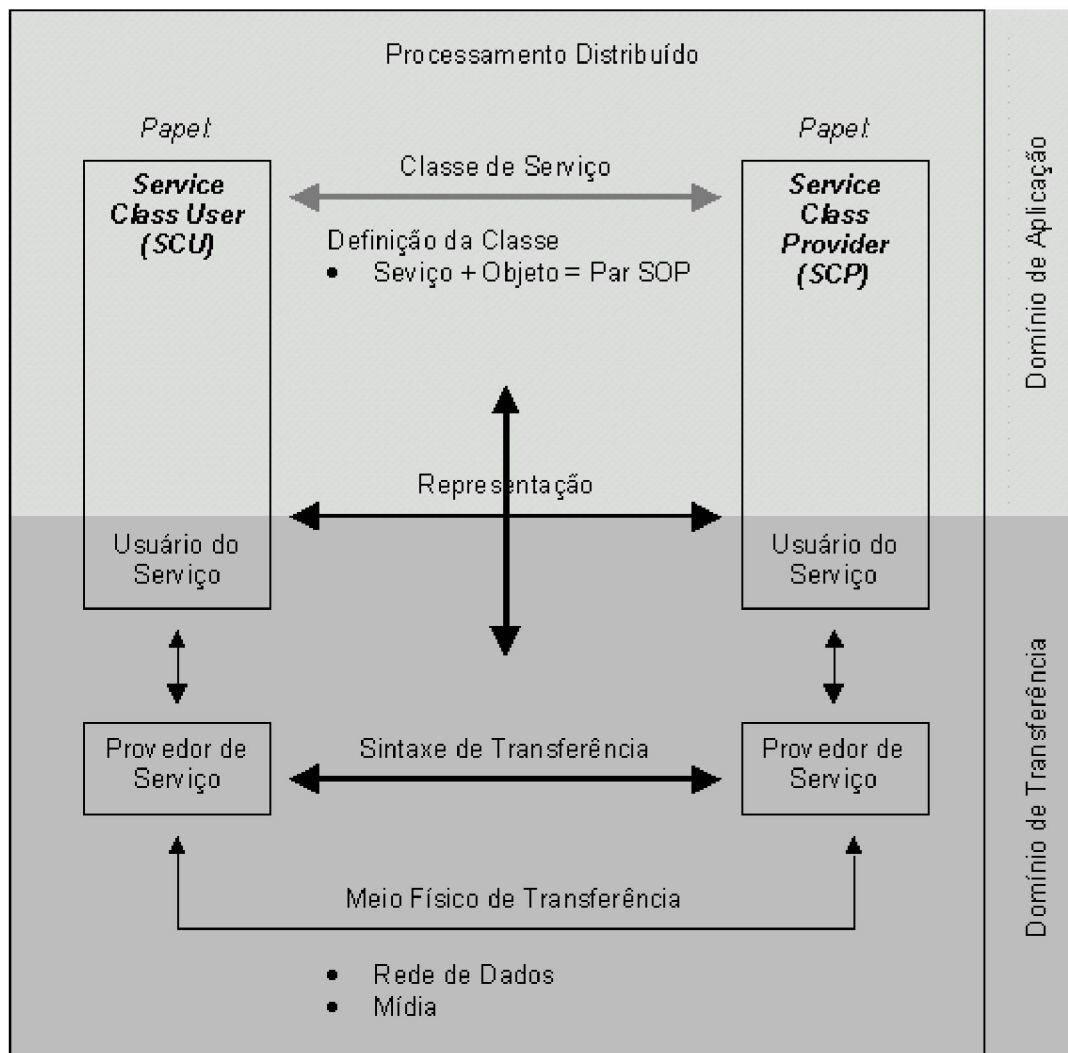


FIGURA 7: Modelo de processos distribuídos DICOM

Ambos os lados devem determinar como a informação é representada em forma de bits/bytes. O provedor de serviço deve determinar em qual formato a informação foi transferida e convertê-la para a representação esperada pelo domínio de aplicação. A representação é conhecida entre os usuários de serviço e provedor em cada lado e entre os provedores de serviço. O meio físico de transferência entre os provedores de serviço pode ser uma rede de computadores ou uma mídia (por exemplo DOR, Fita, CD-RW, etc.). Cada mecanismo tem sua maneira de manipular o conhecimento da representação.

3.2 – Conceitos gerais sobre DICOM

O padrão DICOM cobre parcialmente os conceitos de processamento distribuído discutidos anteriormente [RENET, 1997]. O padrão DICOM utiliza a sua própria terminologia para descrever contexto, relacionamento, etc. Na figura 7 podemos ver os termos DICOM equivalentes aos da figura 6.

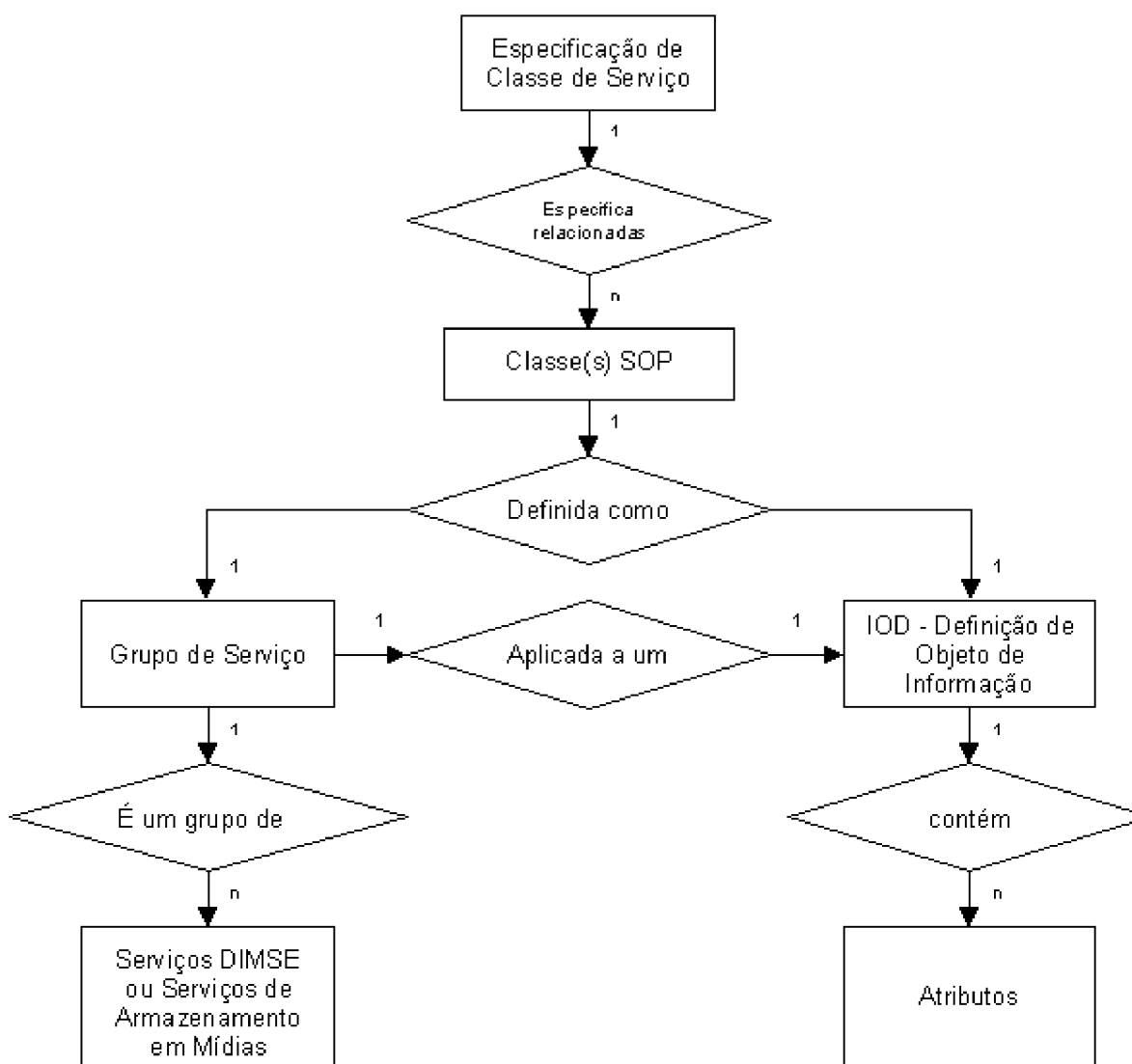


FIGURA 8: Principais estruturas do modelo de informação DICOM

Na figura 8 podemos ver os relacionamentos entre as principais estruturas do modelo de informações DICOM.

3.2.1 - Classes de serviço e classes SOP

O relacionamento entre ambos os parceiros é definida pela descrição de uma Service Class (Classe de Serviço). A classe de serviço descreve explicitamente os papéis que ambos os parceiros fazem. O contexto dos serviços é definido de acordo com a classe de serviço. Em DICOM os papéis que cada aplicação pode assumir são nomeados de SCU (do inglês Service Class User - Usuário da Classe de Serviço), ou cliente, e SCP (do inglês Service Class Provider - Provedor da Classe de Serviço), ou servidor. Não deve-se confundir SCU e SCP com o papel de usuário e provedor de serviço no domínio de transferência mostrado na figura 7.

Uma classe de serviço especifica um grupo de uma ou mais classes SOP (do inglês Service Object Pair - Par Objeto-Serviço) relacionadas a uma função específica, que é aceita por ambas as aplicações, cuja comunicação está em curso. Em cada definição de classe SOP um IOD (do inglês Information Object Definition - Definição de Objeto de Informação) é combinado com um ou mais grupos de serviços. Para cada um desses grupos de serviços os detalhes dos papéis que ambos os parceiros têm de desempenhar são fixados. Mais de uma classe SOP pode existir em uma classe de serviço quando mais de um IOD está envolvido. Uma classe de serviço denota o relacionamento de informação definido em diferentes IOD.

Classes SOP identificam as capacidades de processamento distribuído específicas para uma classe de serviço. Quando parceiros acordam o uso de uma classe SOP, ambos os lados devem assegurar que executarão seu papel como descrito, usando o contexto da classe de serviço em questão. Antes que a transferência de informações tenha início, a identificação da classe SOP é uma questão importante que deve ser resolvida. O mecanismo utilizado depende do tipo de transferência em uso: rede ou mídia. Usando a classe de serviço e outras definições derivadas, parceiros em um ambiente de processamento distribuído funcionam juntos através dos serviços providos pelo domínio de transferência.

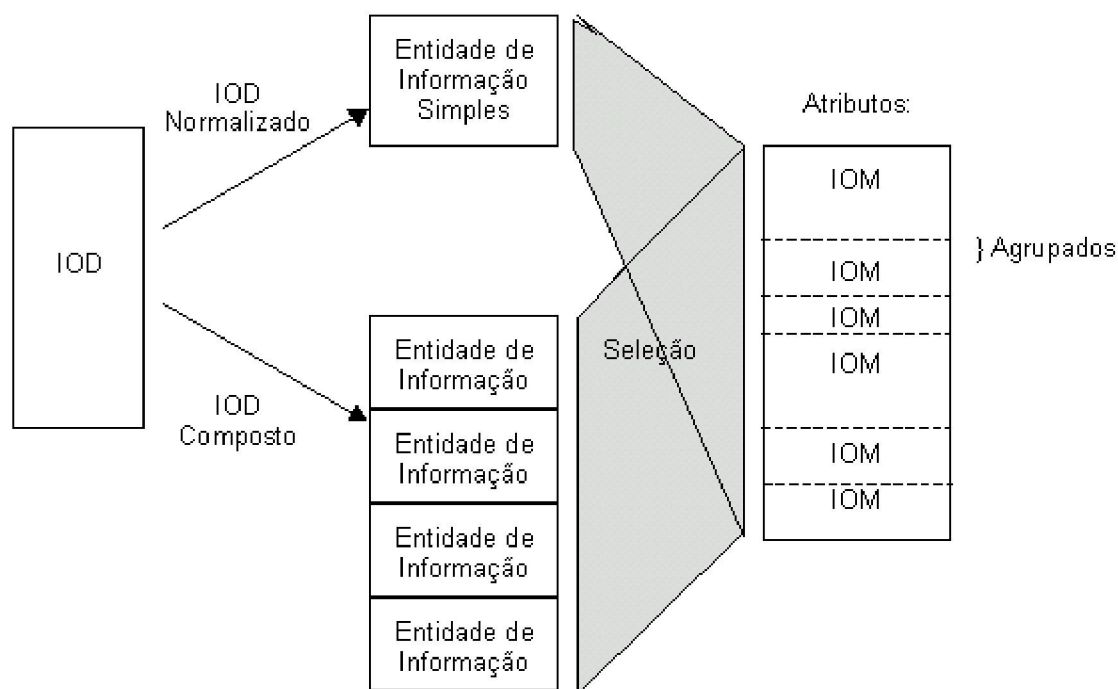


FIGURA 9: Definições de Objetos de Informação e suas Relações

3.2.2 – Definições de objetos de informação

Um IOD (Information Object Definition) é um modelo abstrato de dados orientado a objeto usado para especificar informações de objetos do mundo real. Um IOD provê às aplicações DICOM uma visão em comum sobre a informação a ser trocada. É composto por uma coleção de pedaços de informação relacionados, agrupados em information entities (entidades de informação). A parte relativa a informação de uma classe SOP é definida por IOD's.

Uma entidade de informação consiste de um conjunto de atributos que descrevem um elemento básico do mundo real, como um paciente, um estudo, uma imagem, etc. Atributos que possuem uma relação são agrupados em IOM (Information Object Modules - Módulos de Objeto de Informação). IOM são definidos de maneira que possam ser usados em mais que uma classe IOD. Estes IOM também possuem a vantagem de que a descrição semântica de atributos relacionados podem ser agrupados em conjunto.

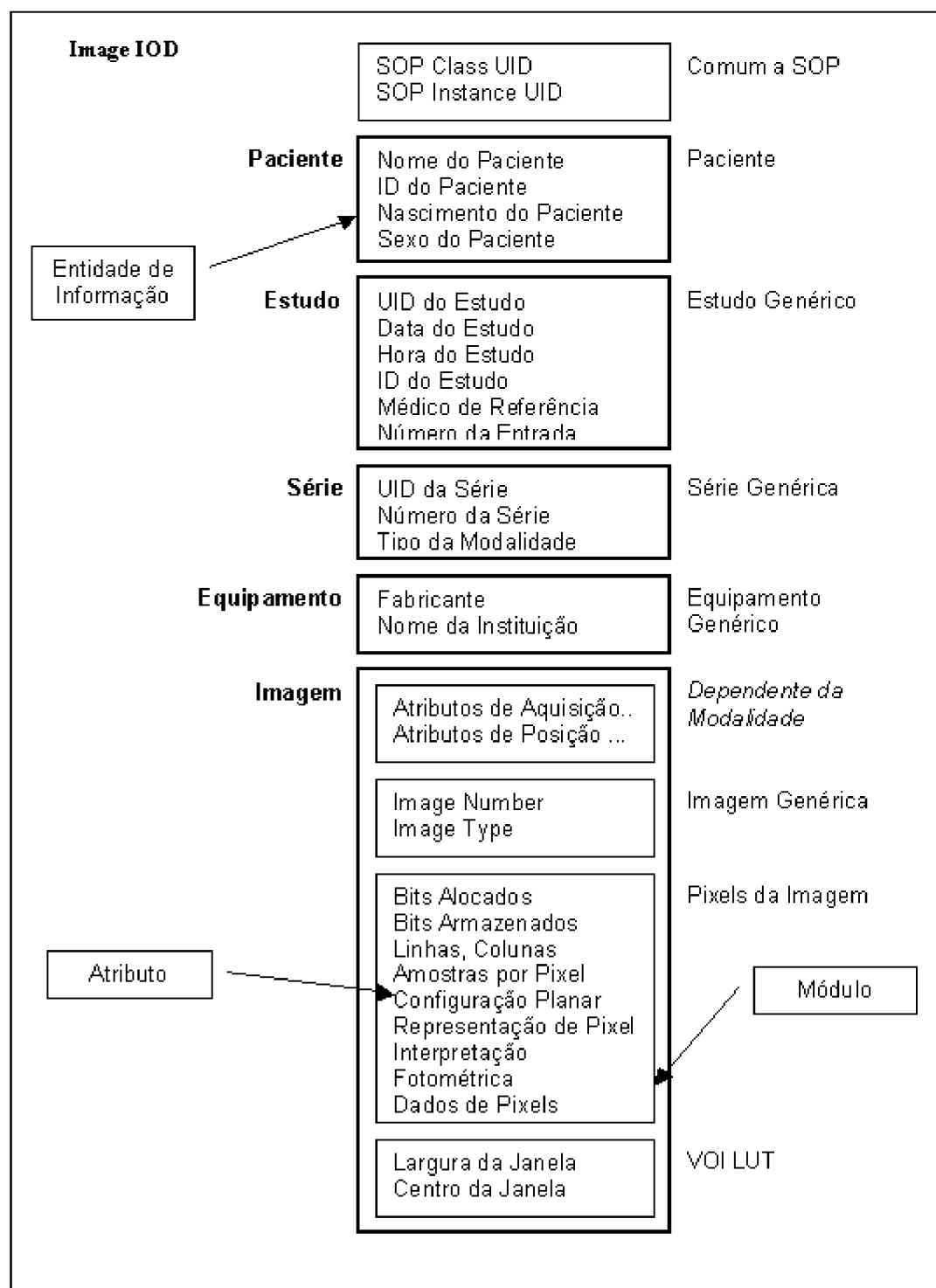


FIGURA 10: Exemplo de um IOD Composto

Na Figura 9 estas relações podem ser observadas. Classes de serviço que implementam funções de gerenciamento (em sua maioria itens simples, como o

gerenciamento de fila de impressão) usam IODs normalizados; aquelas que manipulam seqüências de dados de imagens (uma complexa estrutura de informação) usam IOD compostos.

O relacionamento entre diferentes entidades de informação (estruturalmente) de IOD compostos, é descrito no modelo de informação DICOM (figura 8) como parte da classe de serviço. Com IOD normalizados (apenas uma entidade de informação) não existe a necessidade desta estruturação. Relações para com outras partes da informação são feitas referindo-se àquela informação. Na figura 10 podemos ver um exemplo de IOD Composto, formado por diversas entidades de informação.

3.2.3 - Atributos

Atributos são as entidades de informação básicas e tem de ser descritas em detalhe. Para um atributo as seguintes características são definidas no padrão DICOM:

- **Nome de Atributo** - único (humanamente legível)
- **Identificador do Atributo** - único (legível pelo sistema de informação)
- **Descrição do Atributo** - (semântica)
- **Representação de Valor** - (sintaxe)
- **Multiplicidade de Valor**
- **Classificação de Tipo:** 1, 1C, 2, 2C ou 3 (uso depende do contexto da classe SOP, papel definido na classe de serviço, etc.)

A Classificação de Tipo especifica o uso de atributos relacionados à Classe SOP e aos papéis de SCU ou SCP. Dependendo da situação, cada atributo é forçado a ter um valor (tipo 1) ou forçado a não ter ou ter um valor (tipo 2) ou opcional (tipo 3). Na parte 6 do padrão, é definida para cada atributo um VR (do inglês Value Representation - Representação de Valor). Um VR pode ser: AS (do inglês Age String), PN (do inglês Personal Name), DA (Date), SL (do inglês Signed Long), US (do inglês Unsigned Short), etc. A VM (do inglês Value Multiplicity - Multiplicidade de Valor) de um atributo especifica o número de valores que podem ser codificados para o mesmo. A VM é especificada explicitamente na parte 6 do padrão para cada atributo.

3.2.4 - Elementos de serviço

O padrão DICOM define uma série de elementos de serviço a serem utilizados pelas aplicações. Basicamente temos os serviços de associação e apresentação de dados da OSI Upper Layer Service (camada superior OSI de serviços) definidos na parte PS 3.8 do padrão DICOM e os serviços do protocolo DIMSE (do inglês DICOM Message Service Element).

Os service elements (elementos de serviço) são as operações executadas sobre objetos de informação para uma determinada Classe SOP. O grupo de elementos de serviço faz parte da Classe SOP e é chamado de service group (grupo de serviço).

3.2.4.1 - Serviços de associação

O protocolo DIMSE não provê serviços para o estabelecimento e término de associações entre aplicações. Os serviços de associação estão definidos na parte PS 3.8.

Durante a fase de estabelecimento de uma associação, um usuário de serviços DIMSE deve enviar informações de inicialização usando os parâmetros do serviço da camada superior A-ASSOCIATE, que inclui:

- Contexto da Aplicação
- Requerimentos de Apresentação e Sessão
- Informação de usuário específica ao DIMSE
- Informação de Associação de Aplicação

Os serviços A-RELEASE e A-ABORT definidos na parte PS 3.8 devem ser usados respectivamente para o término de uma associação.

3.2.4.2 - Serviços DIMSE

Existem basicamente dois conjuntos de elementos de serviço:

- DIMSE-C: dão suporte a operações associadas com classes SOP compostas e provêm compatibilidade com as versões anteriores do padrão DICOM.

- DIMSE-N: dão suporte a operações associadas com classes SOP normalizadas e provêm um conjunto estendido de operações e notificações orientadas a objeto.

O grupo de serviço de uma classe SOP é selecionado de uma lista fixa de elementos de serviço DICOM. O contexto descrito pela classe de serviço é limitado quando da utilização de IODs compostos. Cada elemento de serviço tem um significado complexo, por exemplo STORE, FIND, MOVE. Não há relacionamento assumido entre elementos de serviço individuais em uma sequência quando usamos classe de serviço composto.

Porém, classes de serviço usando IODs normalizados têm um contexto muito mais amplo, tal como funções de gerenciamento. Elas usam elementos de serviço primitivos para operações com partes simples da informação: GET, SET, ACTION, etc. A classe de serviço define a relação de uma sequência de requisições primitivas. Com classes de serviço normalizadas ambas as partes seguem o processamento em ambos os lados, usando elementos de serviço para controlá-los. A semântica dos elementos de serviço depende da classe de serviço e da classe SOP na qual estão sendo usados.

Tabela 1: Serviços DIMSE

Nome	Grupo	Tipo
C-STORE	DIMSE-C	Operação
C-GET	DIMSE-C	Operação
C-MOVE	DIMSE-C	Operação
C-FIND	DIMSE-C	Operação
C-ECHO	DIMSE-N	Operação
N-EVENT-REPORT	DIMSE-N	Notificação
N-GET	DIMSE-N	Operação
N-SET	DIMSE-N	Operação
N-ACTION	DIMSE-N	Operação
N-CREATE	DIMSE-N	Operação
N-DELETE	DIMSE-N	Operação

Elementos de Serviço relacionados especificamente a mídias de armazenamento M-WRITE, M-READ, M-DELETE, M-INQUIRE-FILE-SET e M-INQUIRE-FILE definem funções primitivas para manipulação de conjuntos de arquivos.

3.2.5 - Instâncias SOP

Após o acordo sobre quais classes SOP são suportadas (e implicitamente a classe de serviço), e como os papéis de SCU e SCP são divididos, instâncias da classe SOP podem ser transferidas entre as duas partes. Atributos devem ser providos com os valores corretos (semântica) e armazenados dentro da instância SOP como especificado pelas definições de atributos.

A informação coletada será codificada nos formatos definidos pelo padrão DICOM, usando as representações de tag (etiqueta) e valor para criar um DICOM data set (conjunto de dados), no qual cada atributo é codificado em um data element (elemento de dados). Esse conjunto de dados é passado ao provedor de serviço de transferência que garante que o outro lado da associação receba um mesmo conjunto de dados. Diferenças na representação específica ao sistema são levadas em conta durante a transferência, garantindo que a semântica dos valores permaneça intacta. O receptor do conjunto de dados irá decodificá-lo para extrair a informação necessária e agir como acordado pela semântica da classe SOP.

3.2.6 - Identificação

Durante o processo de criação de uma instância SOP é gerada uma identificação para a mesma, como um atributo seu. A identificação é idealizada para uso em sistemas de informação ao invés de uso humano e tem dois aspectos: a identificação de classe e a identificação de instância. Essa identificação tem de ser utilizada em um ambiente de múltiplos fornecedores em diferentes partes do mundo. Para assegurar que cada identificação seja globalmente singular um mecanismo é utilizado para gerar um conjunto de caracteres, conhecido como UID (do inglês Unique Identifier - Identificador Único) como segue:

<raiz>.<sufixo>

A raiz é fornecida por uma autoridade que garante que ninguém mais irá utilizá-la. Esse número é alocado por organizações de padronização para companhias tais como fabricantes, hospitais e clínicas. Ao utilizar uma identificação de sistema única cada sistema terá uma raiz única a nível mundial. O sufixo é criado dinamicamente por cada sistema quando da criação da instância.

3.2.7 - Relações

As UUIDs além de identificarem a classe SOP e a instância SOP, são utilizadas também para identificar uma relação entre as instâncias. Em uma instância composta que contém uma única imagem pertencente a uma série de imagens, a entidade de informação que contém a informação da série será comum para todas essas instâncias. O relacionamento é identificado utilizando-se o mesmo UUID do atributo UUID da instância da série para todas as instâncias compostas pertencentes àquela série. Neste caso apenas uma instância de UUID é requerida, o atributo por si próprio identifica qual tipo de entidade de informação é identificada.

No caso de instâncias normalizadas apenas referências a instâncias fora de si próprias são possíveis; aqui a combinação de uma identificação de classe e instância é requerida. Esse também é o caso quando imagens referem-se a cada uma quando tem uma certa relação.

3.2.8 - Representação de valor

Para cada atributo uma VR (do inglês Value Representation) é definida. Uma representação de valor descreve como um atributo é codificado em um elemento de dados. O conhecimento da representação de valor é compartilhado pelos parceiros na transferência de informações, o processo de codificação e decodificação tem de ter o cuidado de utilizar a VR correta para um atributo (identificada por sua tag, ou etiqueta).

Há duas maneiras de compartilhar essa informação: compartilhando um dicionário de dados que contém todos os possíveis atributos transferidos, ou a inclusão da representação de valor como uma parte do elemento de dados. A última opção aumenta o tráfego de informações durante uma transferência de informações, mas é

muito mais flexível comparado ao uso de um dicionário de dados compartilhado. Especialmente em um ambiente de múltiplos fornecedores, a sincronização de dicionários de dados é muito difícil. Quando a representação de valor está incluída, a mensagem é codificada com uma VR explícita. Quando a representação de valor não está incluída, tem-se uma mensagem codificada com uma VR implícita.

3.2.9 - Sintaxe de transferência

Antes que o conjunto de dados de uma instância SOP possa ser transferido, a maneira pela qual o conjunto de dados está codificado em uma sequência de bytes precisa ser afixada, tanto por acordo, quando uma transferência via rede é utilizada, como por armazenamento junto aos dados no caso de se utilizar mídia. A codificação é especificada por uma sintaxe de transferência. Três aspectos tem de ser definidos pela sintaxe de transferência:

- como é especificada a **VR** (Value Representation);
- o **ordenamento dos bytes** de números de múltiplos bytes (words, longwords): pequeno indiano ou grande indiano (little endian ou big endian);
- No caso de encapsulação (compressão): o **formato de compressão**.

Analogamente às classes SOP, uma sintaxe de transferência é identificada por um UID.

3.3 - Conceitos de rede DICOM

Vimos anteriormente uma série de conceitos do domínio de aplicação definidos pelo padrão DICOM. O domínio de transferência, acrescido de funções de comunicações em rede, terá seus principais conceitos descritos a seguir, passando a ser chamado de domínio de comunicação (figura 11).

3.3.1 - Entidade de aplicação

A maior questão em aplicações de rede distribuídas é como as aplicações podem contar umas às outras. Algumas providências precisam ser tomadas para endereçar a

contraparte e acordar sobre vários tópicos antes que as instâncias SOP possam ser transferidas. Numa rede DICOM parceiros reconhecem uns aos outros através de AE's (Application Entities - Entidades de Aplicação). Uma entidade de aplicação é a parte do processo que lida com a comunicação. Essa contém o usuário de serviço do processo, contendo funções para estabelecer conexões e transferir a informação. Uma entidade de aplicação possui um título, referenciado no padrão DICOM como application title - título da aplicação, que tem de ser usado quando do estabelecimento da comunicação.

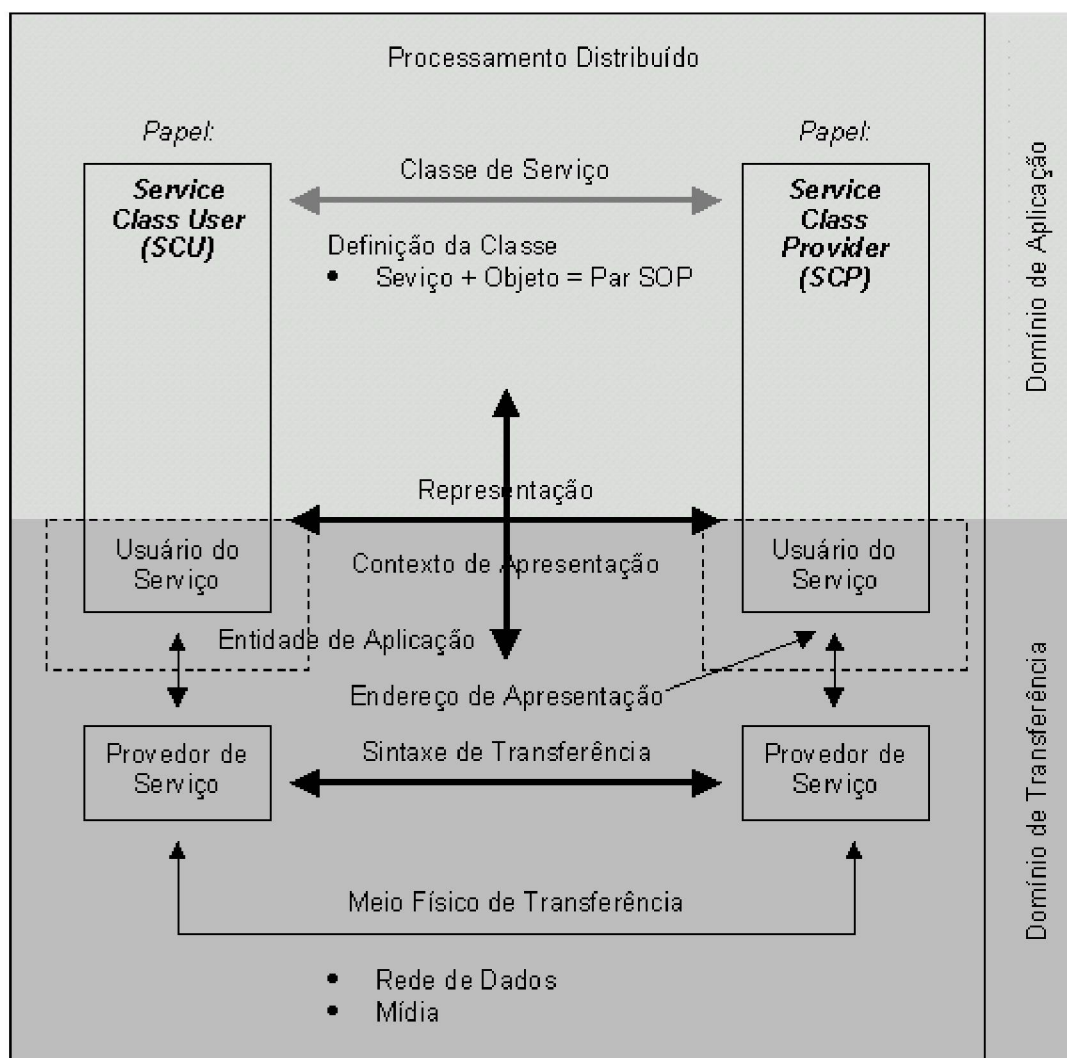


FIGURA 11: DICOM em Rede

3.3.2 - Endereço de apresentação

Application Titles são nomes simbólicos para os processos envolvidos na comunicação. Em uma rede de verdade um endereço de rede deve ser informado. Esse é chamado de endereço de apresentação e aponta para a entidade de aplicação endereçada. O formato do endereço de apresentação depende do protocolo de rede utilizado. A maioria das implementações DICOM utiliza a pilha de protocolos TCP/IP. Neste caso o endereço de apresentação é mapeado para um socket TCP/IP. No caso de se estar utilizando a pilha de protocolo OSI, um OSI Presentation Service Access Point (PSAP) válido deve ser utilizado.

3.3.3 - Negociação de associação

A conexão para transferência de informações entre duas entidades de aplicação é chamada de associação. Para que se possa estabelecer uma associação entre duas entidades de aplicação, algumas questões de comunicação devem ser previamente resolvidas. Primeiramente, o contexto no qual a informação será transferida é negociado. Esse contexto (conhecido como contexto de aplicação) é definido no padrão DICOM e ambos os lados devem acordar em agir em concordância com essa definição de contexto.

3.3.3.1 – Contexto de aplicação

O contexto de aplicação é identificado por um UID e na inicialização de uma associação essa UID é transferida ao parceiro. Comparando essa UID do contexto de aplicação o parceiro pode decidir se é capaz de manipular a requisição de associação e então aceitá-la ou rejeitá-la. Para o padrão DICOM 3 utiliza-se como contexto de aplicação o UID “1.2.840.10008.3.1.1.1”. O parceiro que inicia a negociação de associação propõe as classes SOP que utilizarão, os papéis de SCU e SCP para cada classe SOP proposta e a maneira de representação da informação para cada uma das classes SOP propostas. Dependendo das capacidades do outro lado este aceitará ou rejeitará cada classe SOP individualmente.

3.3.3.2 - Contexto de apresentação

Cada classe SOP que a entidade de aplicação requisitante propõe está inserida em um uma requisição de contexto de apresentação. Cada contexto de apresentação é identificado por um número ímpar entre 1 e 255, chamado de identificador do contexto de apresentação; de um único nome de sintaxe abstrata, que consiste do UID da classe SOP proposta; e de um ou mais UIDs de sintaxes de transferência para cada classe SOP proposta (figura 12).

O parceiro que inicia a negociação propõe todas as sintaxes de transferência que pode manipular para uma determinada classe SOP. De acordo com as classes SOP suportadas pela entidade de aplicação DICOM que recebe o pedido de associação, esta aceitará ou rejeitará cada uma das classes SOP propostas pela entidade de aplicação DICOM que iniciar a negociação. Para cada uma das classes SOP aceitas, uma única sintaxe de transferência deverá ser selecionada, fixando então o contexto de apresentação para as mesmas. Assim, para cada classe SOP aceita, é estabelecido um contexto de apresentação. No contexto de uma associação um número de contextos de apresentação pode existir. O número do contexto de apresentação identifica a classe SOP para que cada transferência de informações possa acontecer.

3.3.3.3 – Notificação de comprimento máximo de PDU

Cada entidade de aplicação DICOM, ao negociar uma nova associação, deverá notificar ao seu parceiro o comprimento máximo em bytes dos itens de dados que suporta. Porém, é permitido que cada parceiro da associação utilize um comprimento máximo de dados diferente do outro (figura 13). Essa negociação é requerida.

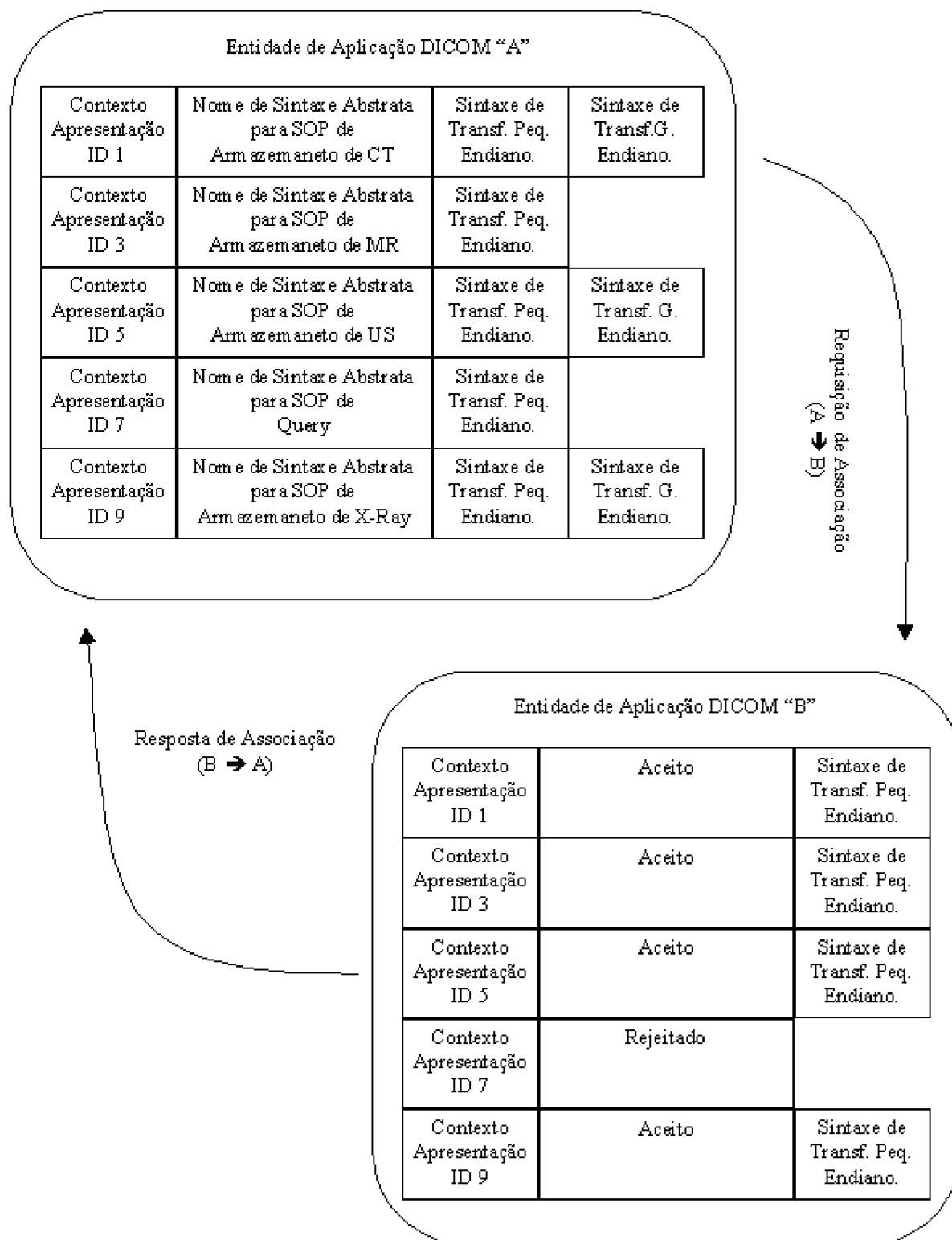


FIGURA 12: Negociação de Contextos de Apresentação

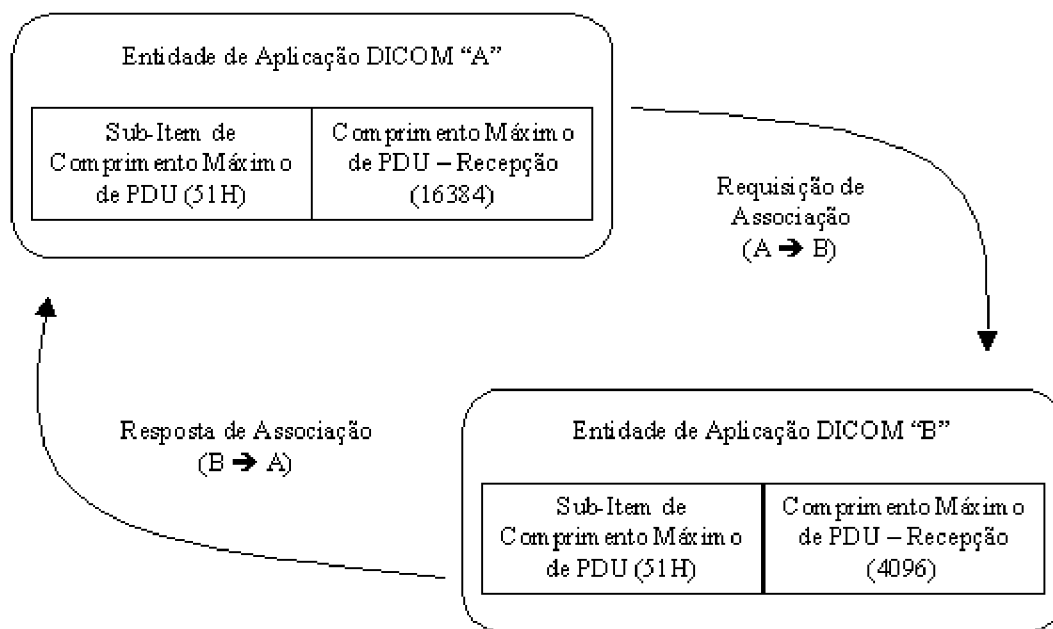


FIGURA 13: Notificação de Comprimento Máximo de PDU

3.3.3.4 - Notificação de identificação de implementação

A notificação de identificação de implementação permite que implementações de entidades de aplicação DICOM em processo de comunicação, reconheçam uma à outra durante a negociação de associação. É uma negociação requerida, pois cada uma das entidades de aplicação envolvidas numa associação conhecerá a outra, e caso ocorram problemas de comunicação, haverá uma identificação não ambígua entre as mesmas.

Cada identificação de implementação está baseada em dois itens:

- UID de classe de implementação (requerido);
- Nome da versão de implementação (opcional).

O UID de classe de implementação identifica de uma maneira única uma classe específica de implementação. Cada entidade de aplicação, que reclamar conformidade com o padrão DICOM, deverá possuir o seu respectivo UID de classe de implementação. A figura 14 demonstra a notificação de UID de classe de implementação.

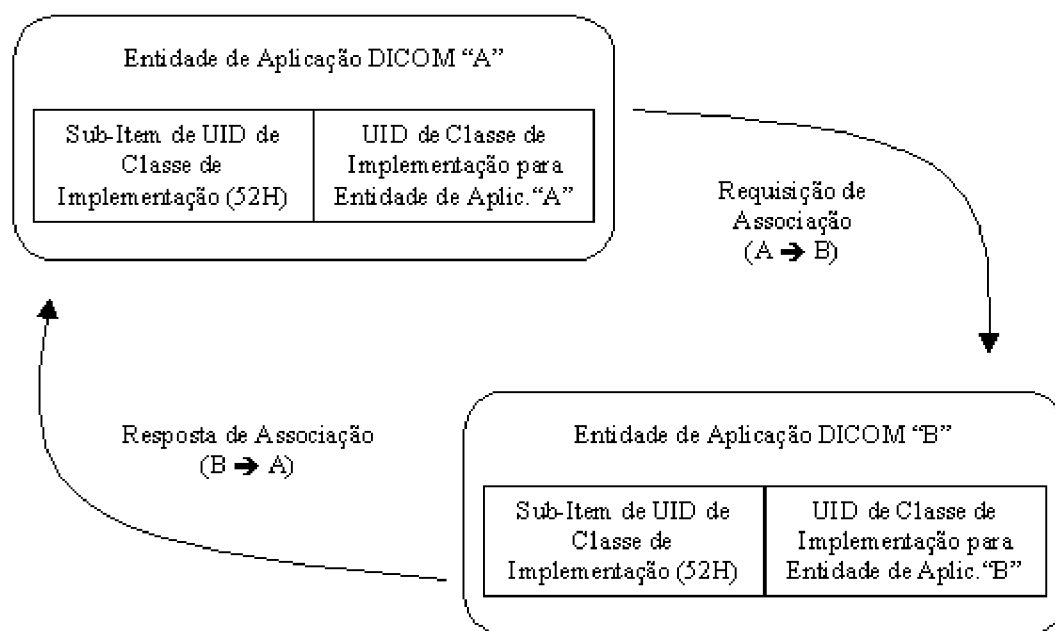


FIGURA 14: Notificação de UID de Classe de Implementação

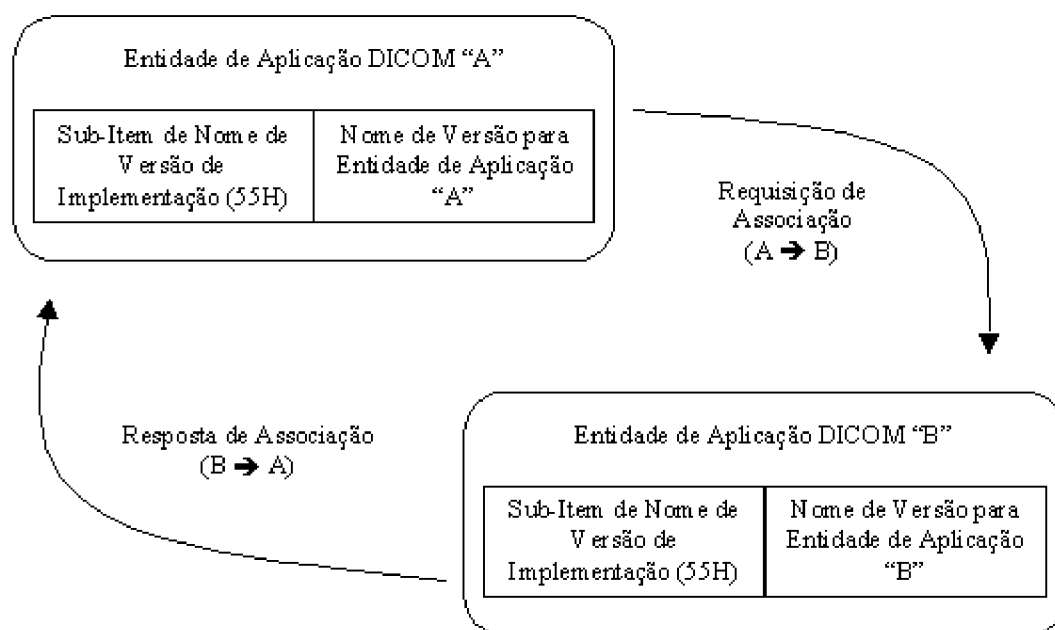


FIGURA 15: Notificação de Nome de Versão de Implementação

Adicionalmente à UID de classe de implementação, existe a opção de se informar um nome de versão de implementação, com até 16 caracteres de comprimento (figura

15). É um ótimo mecanismo para a diferenciação de versões de uma implementação, já que o item de UID de Classe de Implementação não provê uma estrutura adequada para tal.

Após esse processo de negociação ambos os parceiros conhecem as capacidades e limitações do outro. Agora a transferência de informação acontecerá de acordo com a classe de serviço e os papéis da classe SOP definidos para essas classes. Quando a associação não é mais requerida a mesma é encerrada.

3.4 - Tipos de classe de serviço DICOM

O padrão DICOM define uma série de classes de serviço para serem executadas sobre os objetos definidos no padrão. A descrição dos tipos de classes de serviço presentes no padrão DICOM é apresentada a seguir.

3.4.1 - Verification service class

Utilizado apenas para a verificação de uma conexão no domínio de aplicação entre duas entidades de aplicação DICOM. Sua função é similar ao comando `ping` suportado pelo sistema operacional UNIX. Enquanto o comando `ping` verifica os níveis de hardware e rede de uma conexão, a classe de serviços de verificação (C-ECHO) executa esta tarefa sobre uma associação DICOM.

3.4.2 - Store service class

Define uma classe de serviço do domínio de aplicação que possibilita uma simples e fácil transferência de imagens. Isto permite que uma entidade de aplicação DICOM envie imagens à outra. As imagens não são agrupadas em uma entidade lógica particular ou enviadas em alguma ordem definida. O dispositivo receptor (SCP) agrupa e armazena imagens de acordo com a arquitetura do banco de dados nativo. O número de UID pode ser usado para correlações lógicas das imagens.

3.4.3 - Media storage service class

Permite o armazenamento de imagens e informações relacionadas em meios removíveis, tais como fita e/ou discos opto-magnéticos. A informação armazenada pode ser lida em outro dispositivo de armazenamento ou impressa.

3.4.4 - Query/retrieve service class

É uma extensão da classe de serviços de armazenamento. Esta classe de serviço não pretende fornecer um mecanismo de consulta ao banco de dados generalizado como o SQL. Em vez disso, propõe-se a consultas de informações da imagem usando um pequeno conjunto de atributos chave.

Adicionalmente, esta classe de serviços fornece mecanismos para transmitir ou recuperar um determinado conjunto de imagens identificadas. Esta capacidade permite que uma entidade de aplicação DICOM possa recuperar imagens de uma outra entidade de aplicação DICOM, ou requisitar a uma entidade de aplicação DICOM remota que inicie uma transferência de imagens.

3.4.5 - Study content notification service class

Dispositivos DICOM usam esta classe para notificar a existência, conteúdo e localização de origem de uma determinada imagem de um estudo. Utilizado para verificar se todas as imagens em um estudo estão presentes.

3.4.6 - Patient management service class

Esta classe de serviço facilita a criação e acompanhamento dos subconjunto de informações do paciente e suas visitas, auxilia no gerenciamento dos estudos radiológicos deste paciente. A principal função desta classe de serviços é o suporte a entidades de aplicação requisitantes de acesso às informações relacionadas a um determinado paciente com o propósito de obter um ou mais estudos radiológicos deste paciente.

3.4.7 - Study management service class

Esta classe de serviços é análoga à vista no item anterior, facilitando a criação, agendamento, verificação de desempenho e acompanhamento de estudos realizados. Diferindo da maioria das classes de serviço, onde o foco é a imagem, aqui os serviços são centrados nos estudos. Um estudo é definido como uma ou mais séries de imagens. Séries, por sua vez, são um grupo de imagens que são criadas em uma modalidade (CT, NM, ultra-sonografia, etc.).

3.4.8 - Results management service class

Análogo à Patient e Study Management Service Class, esta classe facilita a criação e acompanhamento de resultados e interpretações associadas ao diagnóstico.

3.4.9 - Print management

Esta classe está relacionada à impressão de imagens em dispositivos como filme ou impressora. Fornece um método flexível para formatar a disposição das imagens no filme e para a manipulação de pixels. O usuário dá a formatação esperada e a instância desta classe faz todas as manipulações necessárias.

4 - Suporte para comunicação em rede DICOM

Neste capítulo são apresentados os métodos e definições para a comunicação de dados em redes de computadores de acordo com o a parte PS 3.8 do padrão DICOM.

4.1 - Ambiente de suporte à comunicação em rede

Os serviços de comunicação em rede especificados pelo padrão DICOM na parte 3.8 são um conjunto de serviços genéricos com a finalidade de suportar a comunicação entre entidades de aplicação DICOM. Estes serviços são um subconjunto dos serviços oferecidos pelos OSI Presentation Services (Serviços de Apresentação OSI - ISO 8822) e do OSI Association Control Service Element – ACSE, (Elemento de Serviço de Controle de Associação OSI - ISO 8640). Eles devem ser referenciados como upper layer services ou simplesmente UL services (Serviços da Camada Superior). Os serviços da camada superior DICOM são discutidos na seção 4.2.

Os DICOM UL services permitem o uso de um pilha totalmente conforme com a pilha de protocolos OSI (camadas de 1 a 6 mais o ACSE) a fim de obter comunicações em rede robustas e eficientes, possibilitando a utilização de uma ampla gama de tecnologias de rede baseadas em padrões internacionais, como Ethernet, FDDI, ISDN, X.25 e muitas outras.

Quando essa camada superior de serviços DICOM é provida pelo Upper Layer Protocol for TCP/IP (Protocolo de Camada Superior para TCP/IP), a comunicação de imagens médicas digitais no padrão DICOM poderá ser realizada em uma ampla gama de redes de computadores já instaladas e em funcionamento.

A migração do ambiente de rede TCP/IP para o OSI, sem impacto sobre os elementos de serviço de uma aplicação DICOM, é possível graças à definição de UL services comuns a ambos os ambientes de rede. Esses UL services são também um superconjunto do DICOM Session/Transport/Network Service, definido na parte PS 3.9 do padrão, que suporta a interface ponto-a-ponto (“a interface de 50 pinos”). Essa

característica permite que um dispositivo com uma interface ponto-a-ponto seja conectado em um ambiente de comunicação em rede suportado por OSI ou TCP/IP.

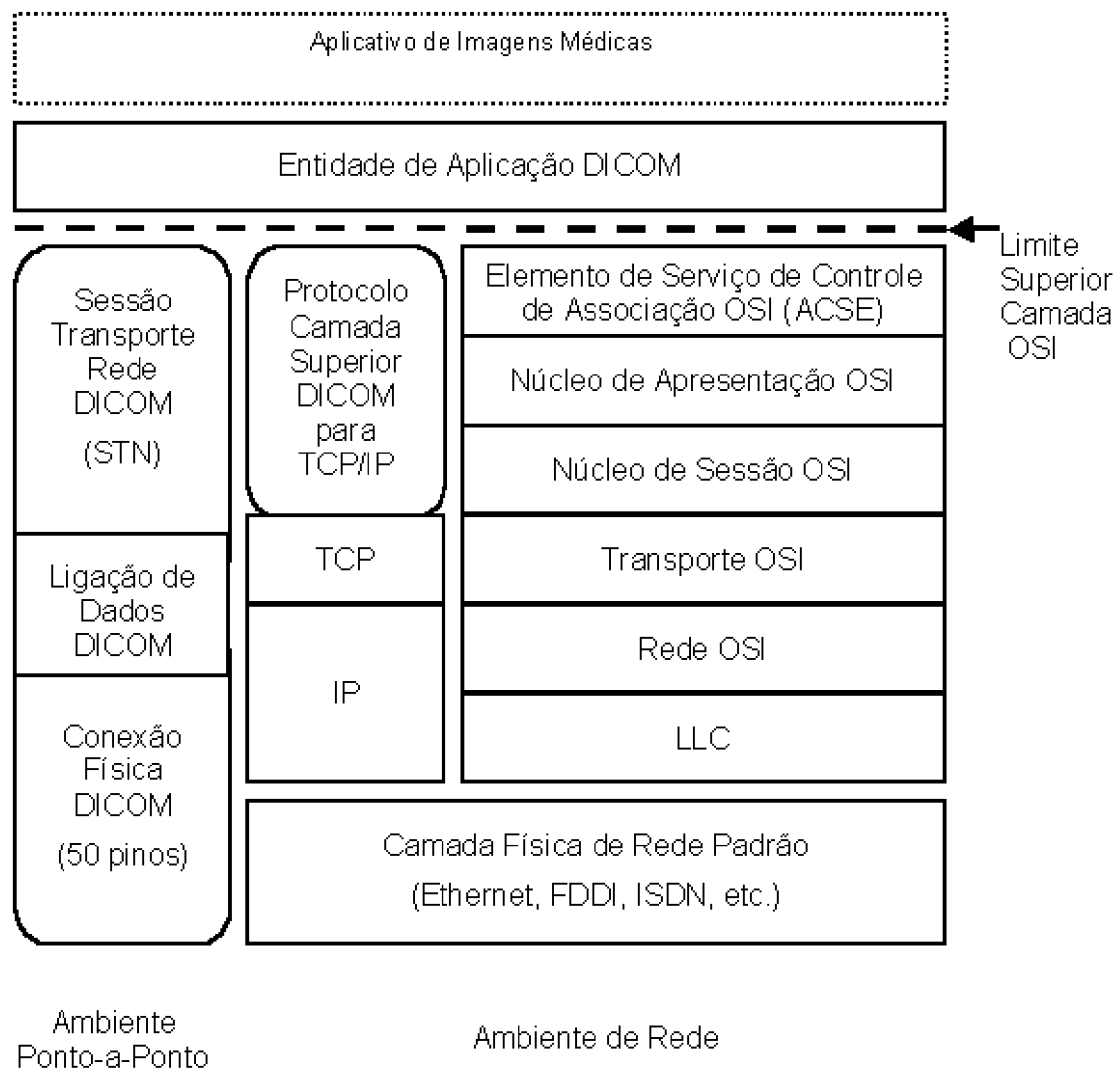


FIGURA 16: Arquitetura de Protocolos do DICOM 3

4.2 - Serviços de camada superior OSI para entidades de aplicação

Os serviços da camada superior são um subconjunto totalmente conforme aos serviços oferecidos pelo ACSE e pela Camada de Apresentação OSI. Os serviços da camada superior são em número de 5 e estão listados na tabela 2.

Tabela 2: Serviços da Camada Superior

SERVIÇO	TIPO
A-ASSOCIATE	Confirmado
A-RELEASE	Confirmado
A-ABORT	Não-confirmado
A-P-ABORT	Iniciado pelo Provedor
P-DATA	Não-confirmado

4.2.1 - Serviço A-ASSOCIATE

Quando do estabelecimento de uma associação entre duas entidades de aplicação DICOM, esta deve ser realizada através das primitivas do serviço A-ASSOCIATE (figura 17). A entidade de aplicação que inicia a associação é chamada de requerente e a que vai aceitar ou não a associação é chamada de requerido.

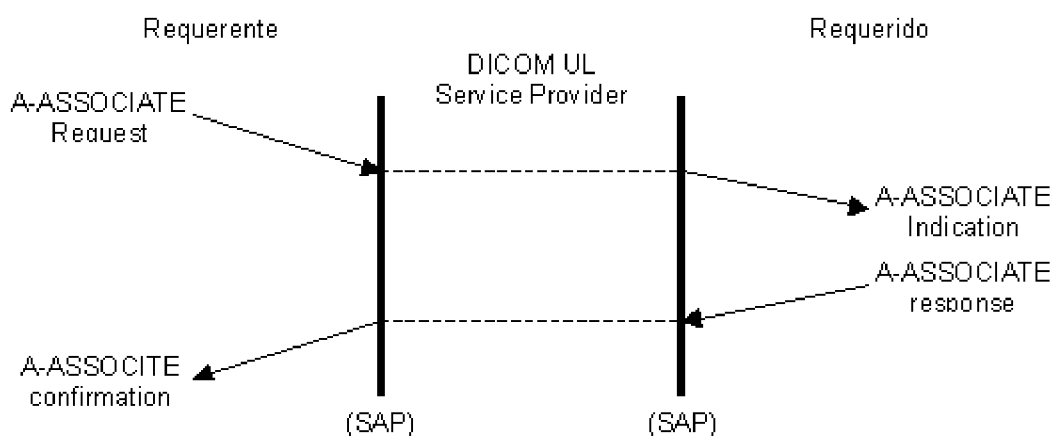


FIGURA 17: ASSOCIATE REQUEST

4.2.1.1 - Procedimentos do serviço A-ASSOCIATE

1. A entidade de aplicação DICOM (que inclui o usuário dos serviços da camada superior) que deseja estabelecer uma associação deve emitir uma primitiva de serviço A-ASSOCIATE request. A entidade de aplicação que está tentando

estabelecer a associação não deve emitir nenhuma outra primitiva de serviço, a não ser A-ABORT request, enquanto não receber a primitiva A-ASSOCIATE confirmation.

2. A camada superior (UL) do provedor do serviço deve emitir uma primitiva de serviço A-ASSOCIATE indication para a entidade de aplicação chamada.
3. A entidade de aplicação chamada pode aceitar ou rejeitar a associação enviando uma primitiva de serviço A-ASSOCIATE response com o parâmetro result apropriado. A camada superior do provedor de serviço deve emitir uma primitiva de serviço A-ASSOCIATE confirmation tendo o mesmo valor para o parâmetro result. O parâmetro result source deve ser atribuído com o valor simbólico "UL-service-user".
4. Se a entidade de aplicação chamada aceitar a associação, a associação está disponível para uso. Ambas as entidades de aplicação podem agora usar qualquer serviço provido pelo contexto de aplicação DICOM que está em efeito, excetuando-se os serviços A-ASSOCIATE.
5. Se a entidade de aplicação chamada rejeitar a associação, a associação não deve ser estabelecida.
6. O provedor de serviço da camada superior pode não ser capaz de suportar a associação que está sendo requerida. Nesta situação, deverá retornar uma primitiva de serviço A-ASSOCIATE confirmation, à entidade de aplicação que emitiu a requisição de associação, com o parâmetro result apropriado (rejected). O parâmetro result source deve ser corretamente atribuído com o valor simbólico "UL service-provider (ACSE related function)" ou "UL service-provider (Presentation Related function)". A primitiva de serviço A-ASSOCIATE indication não deve ser emitida. A associação não será estabelecida.
7. Tanto a entidade de aplicação requerente como a requerida, podem romper o procedimento de serviço A-ASSOCIATE, emitindo uma primitiva de serviço A-ABORT request. A entidade de aplicação remota recebe uma primitiva de serviço A-ABORT indication. A associação não será estabelecida.

4.2.2 - Serviço A-RELEASE

Para que uma associação entre duas entidades de aplicação DICOM seja liberada de maneira correta, ambas devem usar as primitivas de serviço ACSE A-RELEASE request, indication, response e confirmation. A entidade de aplicação que emite a primitiva de serviço A-RELEASE request é a partir de agora chamada de requerente, e a que vai receber a primitiva A-RELEASE indication é a partir de agora chamada de requerida (figura 18).

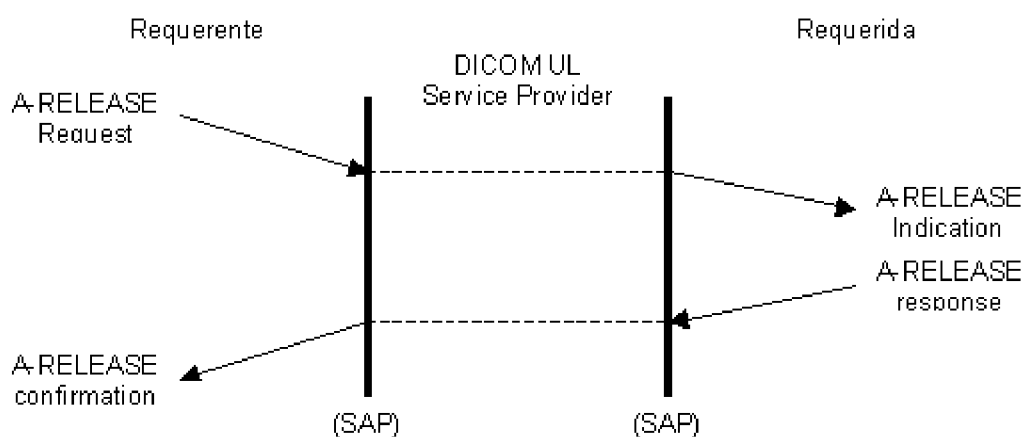


FIGURA 18: ASSOCIATION RELEASE

4.2.2.1 – Procedimentos do serviço A-RELEASE

1. Um usuário de serviços da camada superior que desejar liberar uma associação deve emitir uma primitiva de serviço A-RELEASE request. Após a emissão desta primitiva, este usuário não deverá emitir nenhuma outra primitiva de serviço, a não ser A-ABORT request, enquanto não receber uma primitiva de serviço A-RELEASE confirmation. Mas mesmo assim, ainda poderá receber primitivas de serviço P-DATA indication.
2. O provedor de serviço da camada superior deverá emitir uma primitiva de serviço A-RELEASE indication para a requerida. A requerida não poderá

emitir nenhuma primitiva de serviço diferente de A-RELEASE response, uma primitiva A-ABORT request ou uma primitiva P-DATA request.

3. Para completar o serviço A-RELEASE, a requerida deve responder à primitiva A-RELEASE indication emitindo uma primitiva de serviço A-RELEASE response. Caso aceite liberar a associação, a entidade de aplicação requerida deve sempre responder com uma primitiva de serviço A-RELEASE- response com o valor “affirmative” para o parâmetro result.
4. Após ter emitido um A-RELEASE response, a entidade de aplicação requerida não deve emitir mais nenhuma das primitivas de serviço, incluindo P-DATA requests.
5. O provedor de serviço da camada superior deve emitir uma primitiva de serviço A-RELEASE sempre com “affirmative” como valor do parâmetro result.
6. Um requerente em ambas as entidades de aplicação pode romper o procedimento de serviço A-RELEASE emitindo uma primitiva de serviço A-ABORT request. Quando a entidade de aplicação requerida receber a primitiva A-ABORT indication, a associação será liberada com a possibilidade de perda da informação em trânsito.
7. Uma colisão de procedimento de serviço A-RELEASE acontece quando Requerentes em ambas as entidades de aplicação simultaneamente emitem uma primitiva de serviço A-RELEASE. Nesta situação, ambos os usuários de serviço da camada superior recebem uma primitiva de serviço A-RELEASE indication inesperada. A seguinte seqüência deve ocorrer para completar a liberação normal da associação:
 - a) O requerente da associação deverá emitir um primitiva de serviço A-RELEASE response.
 - b) O requerido na associação espera por uma primitiva de serviço A-RELEASE confirmation de seu ponto. Quando este receber uma, deverá então emitir uma primitiva de serviço A-RELEASE response.

- c) O requerente da associação recebe uma primitiva de serviço A-RELEASE confirmation.

A associação será liberada quando ambos os usuários do serviço ACSE terem recebido um primitiva de serviço A-RELEASE confirmation.

4.2.3 - Serviço A-ABORT

O serviço ACSE A-ABORT pode ser utilizado por um requerente em ambas as entidades de aplicação de uma associação para causar o encerramento anormal desta associação. É um serviço não confirmado. Porém, devido à possibilidade de uma colisão de procedimentos de serviço A-ABORT, o envio da primitiva de A-ABORT indication não é garantido. Caso uma colisão destas venha a ocorrer, ambas as entidades de aplicação envolvidas na associação deverão assumir que esta foi terminada. O encerramento anormal da associação deverá ser através das primitivas de serviço A-ABORT request e A-ABORT indication (figura 19).

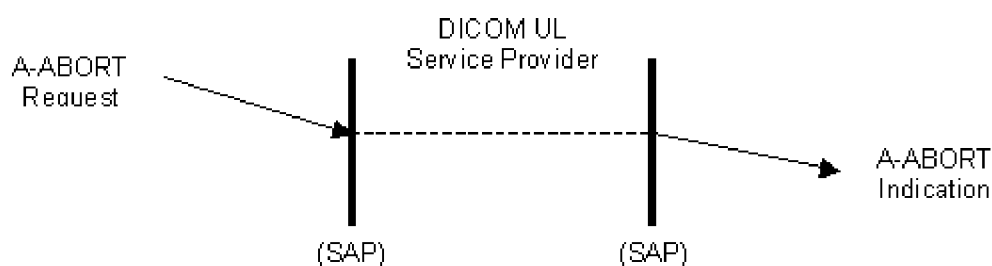


FIGURA 19: Termino anormal da Associação iniciado pelo usuário do serviço

4.2.3.1 - Procedimentos do serviço A-ABORT

1. Quando um serviço A-ABORT é usado, a associação deve ser liberada de maneira anormal e simultaneamente com a liberação anormal da conexão utilizada.
2. Um usuário de serviço de camada superior que deseja liberar a associação de maneira anormal deve emitir uma primitiva de serviço A-ABORT request.

Esse requerente não deve emitir mais nenhuma primitiva de serviço para a associação.

3. O provedor de serviço da camada superior deve emitir uma primitiva de serviço A-ABORT indication para o requerido. O provedor de serviço da camada superior deverá atribuir o valor “UL service user” para o parâmetro Abort Source. A associação e a conexão utilizadas deverão ser liberadas.
4. O provedor de servidor (funções relacionadas a ACSE) pode por si mesmo liberar a associação devido a erros internos. Neste caso, o provedor de serviço da camada superior deve emitir primitivas de serviço A-ABORT indication para ambas as entidades de aplicação. O provedor de serviço da camada superior deve atribuir o valor “UL service-provider” ao parâmetro abort source. O parâmetro user information não deve ser utilizado.

4.2.4 - Serviço A-P-ABORT

O serviço ACSE A-P-ABORT deve ser utilizado pelo provedor de serviço da camada superior para sinalizar a liberação anormal da associação devido a problemas em serviços na camada de apresentação e abaixo. Essa ocorrência indica a possibilidade de perda de informação em trânsito. Um A-P-ABORT é um serviço iniciado pelo provedor (figura 20).

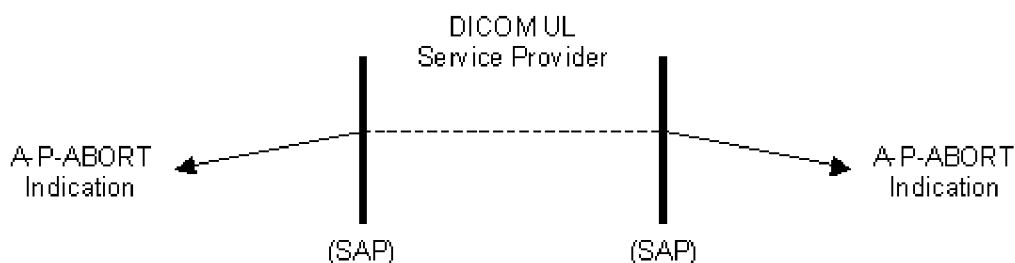


FIGURA 20: Termino anormal da Associação iniciado pelo provedor do serviço

4.2.4.1 - Procedimento do serviço A-P-ABORT

Quando o provedor de serviço da camada superior detecta um erro interno, primitivas de serviço A-P-ABORT devem ser enviadas para ambas as entidades de aplicação. A associação deve ser liberada de maneira anormal. Requerentes em ambas as entidades de aplicação não devem emitir mais nenhuma das primitivas de serviço para a associação.

4.2.5 - Serviço P-DATA

Este serviço pode ser utilizado por ambas as entidades de aplicação de uma associação para o envio de informações de aplicação (por exemplo mensagens DICOM). Mensagens DICOM devem ser enviadas como definido na parte PS 3.7 do padrão. Uma associação provê o envio bidirecional simultâneo de primitivas P-DATA request/indication (figura 21).

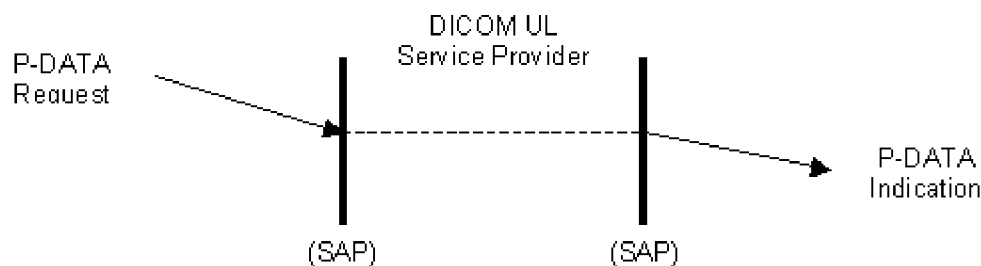


FIGURA 21: Transferência de Dados

4.3 - Protocolo de camada superior DICOM para TCP/IP

Em ambientes de rede que não tenham o suporte ao padrão OSI, pode-se utilizar o DICOM Upper Layer Protocol for TCP/IP (Protocolo de Camada Superior DICOM para TCP/IP). Este é o nosso caso, já que redes com o suporte completo à pilha de protocolos OSI não são muito comuns em nosso país. A implementação tema desta dissertação foi feita utilizando-se do Protocolo de Camada Superior DICOM para TCP/IP.

4.3.1 - Uso do serviço de transporte provido pelo TCP

Existe uma relação de um para um entre uma conexão de transporte TCP e uma associação de camada superior. Para o estabelecimento de uma conexão TCP, uma porta TCP deve ser utilizada para servir como selecionador de transporte. Uma entidade de camada superior DICOM, no ambiente proporcionado pelo TCP/IP, é identificada em um determinado sistema em uma rede pelo número da porta utilizada, que deve ser único no escopo do sistema.

O número da porta TCP/IP de entidades de camada superior DICOM remotas (a bem conhecida porta de número decimal 104, comumente utilizada em aplicações DICOM, ou outro número de porta) deve ser configurável. Entidades de aplicação podem também escolher acessar os serviços de transporte TCP através de uma conexão de transporte segura. A natureza desta conexão de transporte segura é especificada na parte PS 3.15 do padrão DICOM.

4.3.1.1 - Abrindo uma conexão de transporte TCP

Quando uma entidade de camada superior DICOM está para estabelecer uma associação, uma primitiva TRANSPORT CONNECT deve ser enviada ao serviço de transporte TCP (abertura ativa). Assim que a confirmação de conexão de transporte TCP é recebida (abertura completa), um A-ASSOCIATE-RQ PDU deve ser enviado (escrito) na recém estabelecida conexão de transporte.

Ao tornar-se ativa (estado de associação ocioso), uma entidade de camada superior DICOM deve esperar por conexões de transporte TCP em um modo passivo, iniciando um processo de “escuta” (TCP port listening). Quando uma indicação de conexão de transporte TCP é recebida da rede, esta é aceita e um cronômetro chamado de ARTIM (Association Request/Reject/Release Timer) é definido. Qualquer troca de PDUs (leitura/escrita) subsequente deve ser feita como especificado pela Upper Layer State Machine (Máquina de Estados da Camada Superior), incluindo o fim da contagem de tempo pelo cronômetro ARTIM antes do recebimento de um A-ASSOCIATE-RQ PDU.

4.3.1.2 - Transferindo dados através de uma conexão de transporte

A troca de PDUs (leitura e escrita) em uma conexão TCP estabelecida deve seguir as especificações da DICOM Upper Layer Protocol State Machine (Máquina de Estados de Protocolo da Camada Superior DICOM), além da própria estrutura dos PDU's definida na parte PS 3.8 do padrão.

4.3.1.3 - Fechando uma conexão de transporte TCP

Algumas são as situações nas quais uma conexão de transporte TCP são fechadas. Todas são descritas na máquina de estados de protocolo da camada superior DICOM. Alguns casos típicos são:

- a) Após um A-RELEASE-RQ ter sido enviado e o A-RELEASE-RP PDU ser recebido.
- b) Quando uma conexão de transporte foi estabelecida por uma entidade DICOM remota e nenhum A-ASSOCIATE-RQ é recebido antes de o cronômetro ARTIM terminar a contagem de tempo.
- c) Quando um A-ABORT PDU é recebido.
- d) Quando um A-ABORT PDU foi enviado e o cronômetro ARTIM termina a contagem de tempo antes de a conexão de transporte estar fechada.
- e) Quando uma conexão TCP está sendo desconectada pelo provedor do serviço de transporte (por exemplo, falha na rede).
- f) Quando uma conexão TCP está sendo desconectada por uma entidade DICOM remota.

4.3.1.4 - Cronômetro ARTIM

A quantidade de tempo, a ser marcada pelo cronômetro para gerenciar a requisição, rejeição e liberação das associações, deve ser adequada para a maioria das conexões de rede que serão utilizadas pela entidade de camada superior DICOM. Para nossa implementação a quantidade de tempo foi definida como 600 segundos.

4.3.2 - Protocolo de camada superior para a máquina de estados TCP/IP

A máquina de estados TCP/IP pode estar em treze diferentes estados, que são divididos em cinco categorias. As cinco tabelas a seguir definem estes estados.

Tabela 3: Nenhuma Associação

Estado	Definição
Sta 1	Idle

Tabela 4: Estabelecimento de Associação

Estado	Definição
Sta 2	Conexão de Transporte Aberta (aguardando A-ASSOCIATE-RQ PDU)
Sta 3	Aguardando primitiva A-ASSOCIATE response local (do usuário local)
Sta 4	Aguardando término da abertura da conexão de transporte (serviço transporte local)
Sta 5	Aguardando A-ASSOCIATE-AC ou A-ASSOCIATE-RJ PDU

Tabela 5: Transferência de Dados

Estado	Definição
Sta 6	Associação estabelecida e pronta para transferência de dados

Tabela 6: Liberação de Associação

Estado	Definição
Sta 7	Aguardando A-RELEASE-RP PDU
Sta 8	Aguardando primitiva A-RELEASE release local (do usuário local)
Sta 9	Colisão de Liberação no lado do Requerente; aguardando A-RELEASE response (do usuário local)
Sta 10	Colisão de Liberação no lado do Requerido; aguardando A-RELEASE-RP PDU
Sta 11	Colisão de Liberação no lado Requerente; aguardando A-RELEASE-RP PDU
Sta 12	Colisão de Liberação no lado Requerido; aguardando primitiva A-RELEASE response (do usuário local)

Tabela 7: Fechamento da conexão de transporte

Estado	Definição
Sta 13	Esperando indicação de fechamento da conexão de transporte (associação não existe mais)

O protocolo de camada superior DICOM para máquina de estados TCP/IP também define uma série de ações, que devem ser implementadas para responder aos eventos que podem acontecer durante o espaço de tempo em que a máquina de estados TCP/IP estiver ativa.

A definição destas ações e a transição de estados da máquina de estados podem ser vistas nas tabelas 8 a 11.

Tabela 8: Ações Relacionadas ao Estabelecimento de uma Associação

Ação	Definição
AE-1	Emitir primitiva TRANSPORT CONNECT para o serviço de transporte local Próximo estado é Sta 4
AE-2	Enviar A-ASSOCIATE-RQ PDU Próximo estado é Sta 5
AE-3	Emitir primitiva A-ASSOCIATE confirmation (accept) Próximo estado é Sta 6
AE-4	Emitir primitiva A-ASSOCIATE confirmation (reject) e fechar conexão de transporte Próximo estado é Sta 1
AE-5	Emitir primitiva Transport Connection response, iniciar cronômetro ARTIM Próximo estado é Sta 2
AE-6	Parar cronômetro ARTIM, se A-ASSOCIATE-RQ aceitável pelo provedor do serviço: Emitir primitiva A-ASSOCIATE indication Próximo estado é Sta 3 senão: Emitir A-ASSOCIATE-RJ-PDU e iniciar cronômetro ARTIM Próximo estado é Sta 13
AE-7	Enviar A-ASSOCIATE-AC PDU Próximo estado é Sta 6
AE-8	Enviar A-ASSOCIATE-RJ PDU e iniciar cronômetro ARTIM Próximo estado é Sta 13

Tabela 9: Ações Relacionadas à Transferência de Dados

Ação	Definição
DT-1	Enviar P-DATA-TF PDU Próximo estado é Sta 6
DT-2	Enviar primitiva P-DATA indication Próximo estado é Sta 6

Tabela 10: Ações Relacionadas à Liberação de Associação

Ação	Definição
AR-1	Enviar A-RELEASE-RQ PDU Próximo estado é Sta 7
AR-2	Emitir primitiva A-RELEASE indication Próximo estado é Sta 8
AR-3	Emitir primitiva A-RELEASE confirmation, e fechar conexão de transporte Próximo estado é Sta 1
AR-4	Emitir A-RELEASE-RP PDU e iniciar cronômetro ARTIM Próximo estado é Sta 13
AR-5	Parar cronômetro ARTIM Próximo estado é Sta 1
AR-6	Emitir primitiva P-DATA indication Próximo estado é Sta 7
AR-7	Enviar P-DATA-TF PDU Próximo estado é Sta 8
AR-8	Emitir primitiva A-RELEASE indication (colisão de liberação): <ul style="list-style-type: none"> - se requerente da associação, próximo estado é Sta 9 - senão, próximo estado é Sta10
AR-9	Enviar A-RELEASE-RP PDU Próximo estado é Sta 11
AR-10	Emitir primitiva A-RELEASE confirmation Próximo estado é Sta 12

Tabela 11: Ações Relacionadas à Término Anormal (ABORT) de Associação

Ação	Definição
AA-1	Enviar A-ABORT PDU (service user source) e iniciar (ou reiniciar se já iniciado) o cronômetro ARTIM; Próximo estado é Sta 13
AA-2	Parar cronômetro ARTIM se estiver rodando. Fechar Conexão de Transporte Próximo estado é Sta 1
AA-3	Se (usuário do serviço iniciou processo de encerramento anormal) Emitir primitiva A-ABORT indication e fechar conexão de transporte Senão (foi o provedor do serviço que iniciou) Emitir primitiva A-P-ABORT indication e fechar conexão de transporte
AA-4	Emitir primitiva A-P-ABORT indication; Próximo estado é Sta1
AA-5	Parar cronômetro ARTIM; próximo estado é Sta 1
AA-6	Ignorar PDU, Próximo estado é Sta 13
AA-7	Enviar A-ABORT PDU, Próximo estado é Sta 13
AA-8	Enviar A-ABORT PDU (service provider source); emitir uma A-P-ABORT indication e iniciar o cronômetro ARTIM; Próximo estado é Sta 13

A tabela 12 define como se dá a transição de estados da máquina de estados TCP/IP, de acordo com os eventos que ocorrem tanto para a entidade de aplicação que inicia a associação como para que a recebe.

Tabela 12: Transição de Estados do Protocolo de Camada Superior DICOM

[illegible]

4.3.3 - Protocolo da camada superior DICOM para estrutura das unidades de dados TCP/IP

PDU (do inglês Protocol Data Units - Unidades de Dados de Protocolo), são os formatos de mensagens trocadas entre entidades pertencentes a uma camada. Um PDU deve consistir de informação de controle de protocolo e dados de usuário. PDU são construídas com campos de tamanho fixo, seguidos de campos de tamanho variável opcionais, que contém um ou mais itens ou sub-itens.

Os itens de tipos não reconhecidos devem ser ignorados. Itens devem aparecer em uma ordem crescente de acordo com seus tipos de item. Várias instâncias de um mesmo item podem ser aceitáveis ou não como especificado para cada tipo de item. O protocolo de camada superior DICOM consiste de sete PDUs:

- a) A-ASSOCIATE-RQ PDU
- b) A-ASSOCIATE-AC PDU
- c) A-ASSOCIATE-RJ PDU
- d) P-DATA-TF PDU
- e) A-RELEASE-RQ PDU
- f) A-RELEASE-RP PDU
- g) A-ABORT PDU

A estrutura dos sete tipos de PDUs bem como a de seus sub-itens está definida na parte PS 3.8 do padrão DICOM, itens 9.3.2 a 9.3.8.

5 - Implementando o servidor de imagens médicas digitais

Neste capítulo são descritos os métodos, materiais e equipamentos utilizados na implementação do protótipo de Servidor de Imagens Médicas Digitais DICOM, bem como os resultados obtidos e os testes efetuados.

5.1 - Cyclops DICOM editor

Atualmente, no projeto Cyclops, utiliza-se um cliente DICOM chamado de “DICOM Editor” (figura 22), capaz de carregar imagens médicas digitais no formato DICOM de servidores, como, por exemplo, o CTN, através de uma rede de computadores do tipo TCP/IP. Este cliente DICOM (SCU) suporta os IODs para as modalidades de Ressonância Magnética (MR), Tomografia Computadorizada (CT), Ultra-Som (US), Medicina Nuclear (NM), Raios X (X-RAY) e Secondary Capture (SC). Este, bem como os demais softwares desenvolvidos dentro do projeto, foi escrito com a linguagem de programação VisualWorks Smalltalk.

O DICOM Editor permite o acesso às informações de todos os pacientes armazenados em um servidor de imagens configurado para tal. Após selecionar um determinado paciente, são carregadas as informações de todos os estudos do mesmo. Da mesma maneira, todas as séries são carregadas a partir da escolha de um determinado estudo.

Existe ainda um bloco de notas que permite visualizar atributos do paciente, do estudo e da série em questão. Estes atributos podem ser alterados e estas mudanças podem ser gravadas junto às imagens.

Quando da carga de uma série, todas as imagens pertencentes a esta são carregadas ao comando do usuário, sendo exibidas na parte inferior da tela. Caso o usuário deseje visualizar as imagens, deverá abrir o Cyclops DICOM Series Editor (figura 23). Além da visualização, estão disponíveis outras operações sobre as imagens para o usuário, tais como a mensuração de distâncias entre pontos na imagem, mudança dos parâmetros da janela de visualização da imagem, etc.

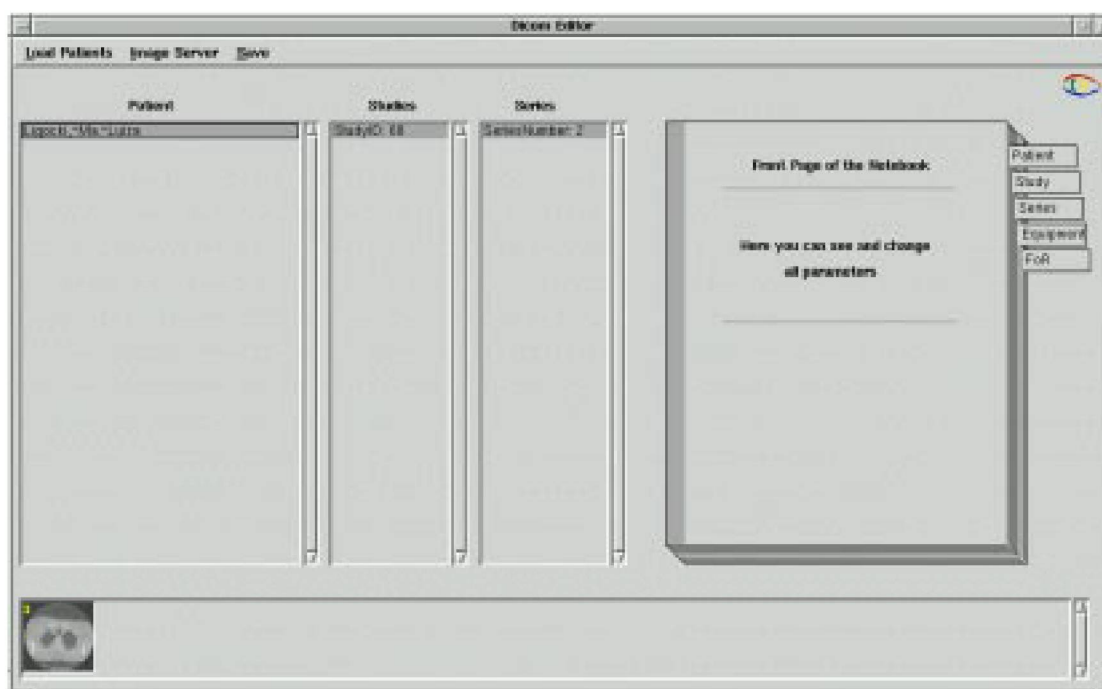


FIGURA 22: Tela do Cyclops DICOM Editor

5.2 - Análise em função dos objetivos

Para que possamos ter um protótipo funcional de um Servidor de Imagens Médicas Digitais DICOM, os objetivos específicos definidos no item 1.3.2 desta dissertação devem todos ser satisfeitos. A análise de cada um dos objetivos propostos e a solução utilizada para atingi-los estão abaixo enumeradas.

5.2.1 - Primeiro objetivo específico

O objetivo a ser atingido é a implementação em VisualWorks Smalltalk da camada de serviços denominada pelo padrão DICOM de “DICOM Upper Layer Protocol for TCP/IP”. A idéia inicial seria utilizar o suporte a tal protocolo já implementado no Cyclops DICOM Editor, através de técnicas de reutilização de código orientado a objeto. Porém, uma análise do código já implementado neste aplicativo, revelou que o mesmo não havia sido modelado/implementado como definido pelo padrão, a fim de dar suporte às funcionalidades necessárias em tal camada para a operação do software como um DICOM SCP - “Service Class Provider”.

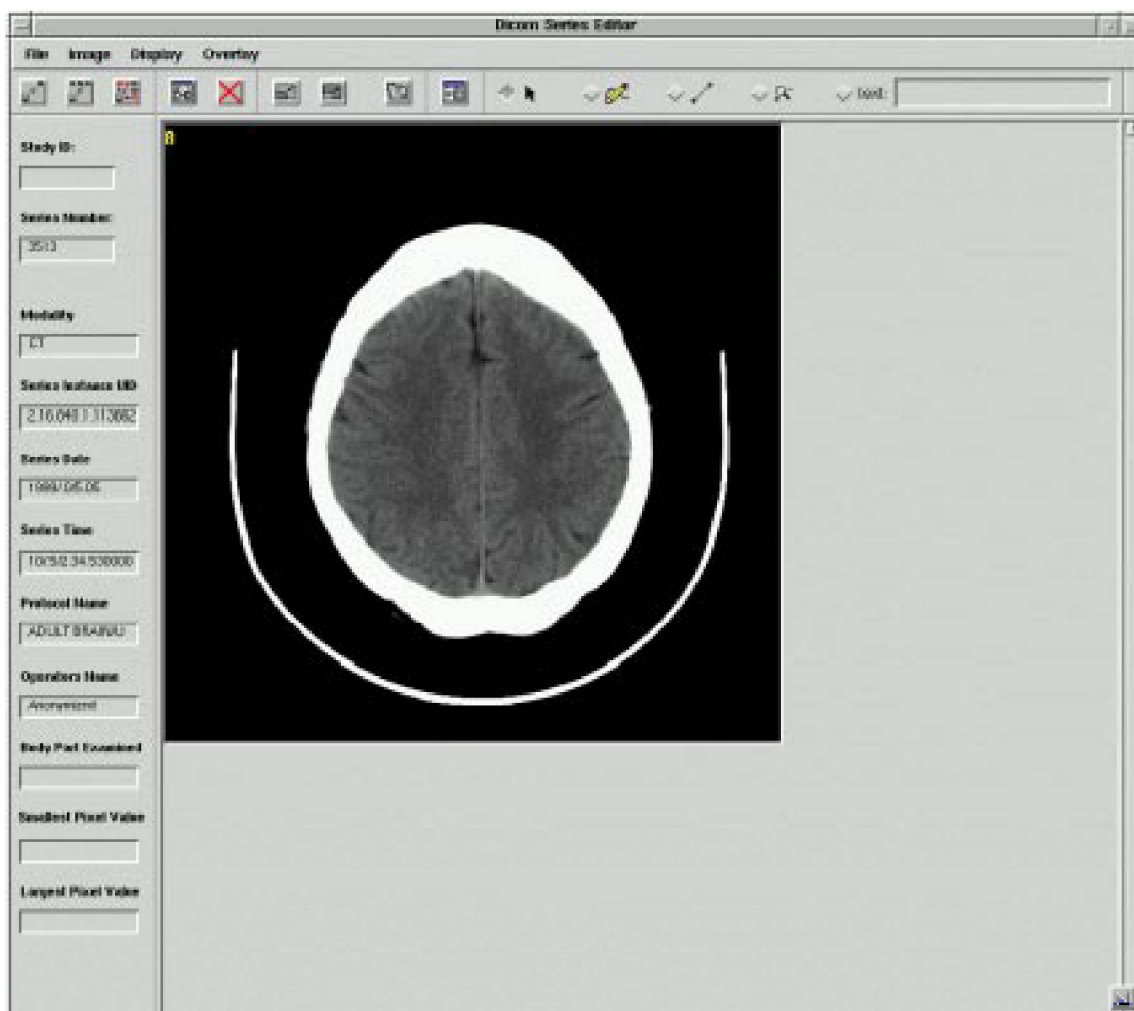


FIGURA 23: Tela do Cyclops DICOM Series Editor

O padrão diz que para esta camada de protocolo DICOM, deve-se implementar uma máquina de estados como a definida na parte PS 3.8. Além disso, deve-se implementar um mecanismo para a detecção dos eventos previstos no padrão, além das ações (também previstas no padrão) e mudanças de estados que são executadas em resposta aos eventos detectados, conforme pode ser visto no item 4.3.2. desta dissertação.

Visto que a implementação já existente estava em desacordo com o que define o padrão, impossibilitando o seu uso no protótipo em desenvolvimento, resolveu-se que uma nova implementação se fazia necessária. A nova implementação foi iniciada do zero, sem nenhuma reutilização de código já existente.

5.2.2 - Segundo objetivo específico

Este objetivo específico visa a implementação das funções e serviços pertinentes à camada “DICOM Application Entity”, para a comunicação conforme especificado no padrão com a camada “DICOM Upper Layer Protocol for TCP/IP”, além do suporte necessário (funções e serviços) às aplicações médicas que já fazem uso de serviços DICOM, em VisualWorks Smalltalk, no contexto do Projeto Cyclops. Verificou-se na implementação do Cyclops DICOM Editor que tal camada não estava implementada de acordo com as definições do padrão DICOM. Na verdade, as camadas “DICOM Application Entity” e “DICOM Upper Layer Protocol for TCP/IP” estão todas misturadas em uma única classe de objetos Smalltalk com o nome de `CyclopsDicomConnection`, o que impossibilitaria uma implementação limpa destas camadas no protótipo em desenvolvimento.

Resolveu-se então também codificar a partir do zero a implementação das funções e serviços desta camada DICOM.

5.2.3 - Terceiro objetivo específico

Este objetivo visa a implementação do que é definido pelo padrão DICOM como “DICOM Model of the Real World”. Mais especificamente; das IODs de imagens e outras modalidades, além das entidades de informação de pacientes, estudos, séries, etc., e seus inter-relacionamentos. A implementação já existente no Projeto Cyclops deste “modelo de mundo” pôde ser reutilizada para os fins de desenvolvimento deste protótipo de software. Foram reutilizados o IOD CT Image e todas as IE (Entidades de Informação) que compõe este IDO neste protótipo.

5.2.4 - Quarto objetivo específico

Implementar a classe de serviços de armazenamento, denominada pelo padrão de “Dicom Image Storage Services”. O Cyclops DICOM Editor já implementa esta classe de serviços, que é utilizada basicamente para a transferência de imagens e informações correlatas entre Entidades de Aplicação DICOM, porém no papel de SCU. Esta implementação já existente teve de ser adaptada às novas classes de objetos

correspondentes às camadas “DICOM Upper Layer Protocol for TCP/IP” e “Application Entity” implementadas. Adicionalmente, em face à sua simplicidade, a classe de serviços C-ECHO foi também devidamente programada. A classe de serviço C-ECHO é descrita conceitualmente no item 3.4.1 desta dissertação.

5.3 - Resultados obtidos

O protótipo de Servidor de Imagens Médicas Digitais DICOM que foi implementado tem suas funcionalidades descritas nos itens a seguir.

5.3.1 - Estabelecimento de associações com outras DICOM AE

Neste quesito, o protótipo implementado está em conformidade com os protocolos para negociação de associações com outras Entidades de Aplicação DICOM especificados nas partes PS 3.8 e PS 3.7 do padrão.

Como as únicas classes de serviço suportadas pelo protótipo implementado são a “CT Image Storage” (armazenamento de imagens de tomografia computadorizada) e a C-ECHO (verificação) só é possível estabelecer uma associação, com esta Entidade de Aplicação DICOM prototípica, especificamente para o fornecimento destas classes de serviço.

Tentativas de se estabelecer conexões para o fornecimento de outras classes de serviço são, como definido pelo padrão, formalmente rejeitadas.

5.3.2 - Transferência de imagens de Tomografia Computadorizada

Para perfazer os serviços da classe de armazenamento de imagens de tomografia computadorizada, a transferência das informações codificadas em DICOM destas imagens digitais é plenamente suportada. Estas informações codificadas de acordo com o padrão DICOM são decodificadas e instâncias dos objetos em Smalltalk que representam um CT Image IOD são criadas, sendo então estas visualizadas utilizando-se do Cyclops DICOM Series Editor.

A funcionalidade do serviço de armazenamento de imagens de tomografia computadorizada destes serviços foi testada utilizando-se os seguintes programas:

- storescu [anexo 2] - utilitário para o envio de arquivos DICOM para entidades de aplicação DICOM via rede TCP/IP, integrante do pacote DCMTK fornecido pela OFFIS [OFFIS];
- send_image [anexo 3] - utilitário para o envio de arquivos DICOM para entidades de aplicação DICOM via rede TCP/IP, integrante do CTN [MALLINCKRODT INSTITUTE OF RADIOLOGY];
- eFilm – “estação de trabalho” radiológica para Windows [EFILM MEDICAL INC.].

5.3.3 – Verification Service Class

O servidor DICOM desenvolvido é capaz de atuar no papel de SCP para a classe de serviços C-ECHO. A única finalidade desta classe é a verificação do funcionamento da conexão (associação) entre duas entidades de aplicação DICOM. A funcionalidade deste serviço no servidor DICOM implementado foi testada utilizando-se os seguintes programas:

- echoscu [anexo 4] - utilitário para a verificação de conexão entre entidades de aplicação DICOM via rede TCP/IP, integrante do pacote DCMTK fornecido pela OFFIS [OFFIS];
- dicom_echo [anexo 5] - utilitário para a verificação de conexão entre entidades de aplicação DICOM via rede TCP/IP, integrante do CTN [MALLINCKRODT INSTITUTE OF RADIOLOGY];

5.4 - Discussão

Todos os resultados obtidos foram de testes realizados com os pacotes de software DCMTK [OFFIS] e CTN [MALLINCKRODT INSTITUTE OF RADIOLOGY], especialmente desenvolvidos para testar funcionalidades em Entidades de Aplicação DICOM. Também foram realizados testes de transferências de imagens

de tomografia computadorizada com eFilm [EFILM MEDICAL INC.]. A plataforma operacional foi o Linux, distribuição Slackware 7.0 com kernel 2.2.18. A arquitetura de rede utilizada foi a TCP/IP. A versão do VisualWorks Smalltalk utilizada foi a 5i.2.

O fato é que o protótipo desenvolvido, no que diz respeito a um Servidor de Imagens Médicas Digitais DICOM, possui muito poucos serviços a oferecer. Porém, a parte mais complexa pertinente a uma Entidade de Aplicação DICOM, a estrutura de camadas DICOM e seus protocolos de serviço, está modelada e implementada em plena concordância com o definido pelo padrão. A extensão das funcionalidades deste protótipo como Servidor DICOM depende da implementação das demais Classes de Serviço DICOM que se fizerem necessárias.

6 - Conclusões

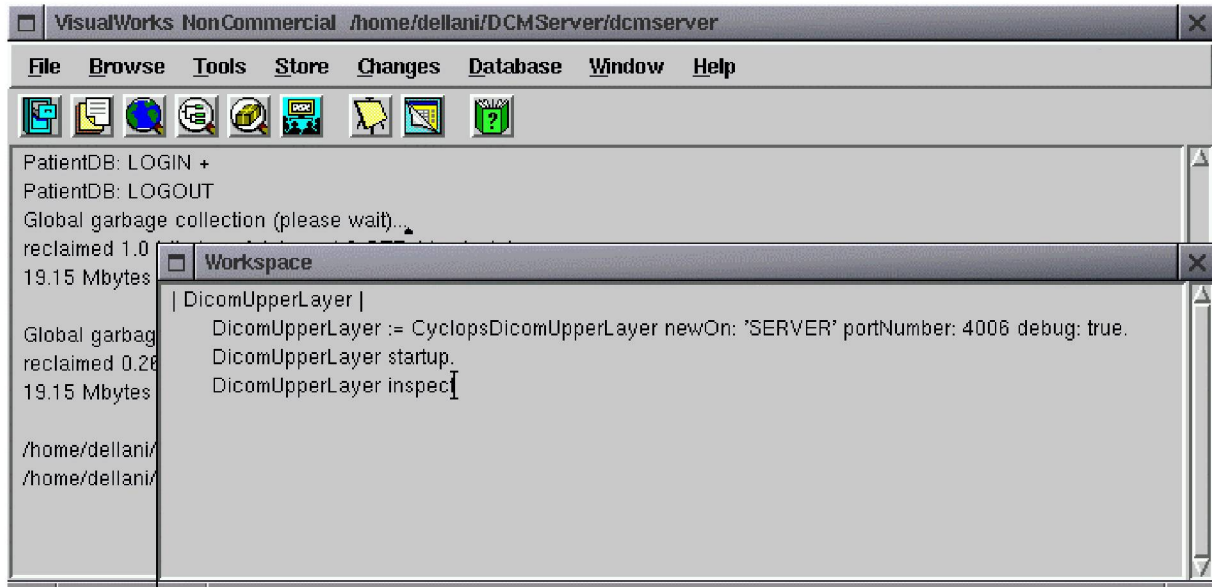
A implementação de um software como um Servidor de Imagens Médicas Digitais DICOM não é tarefa para ser realizada por uma pessoa apenas. A extensão das classes de serviços suportadas pelo padrão DICOM certamente dará muito trabalho de implementação a este indivíduo. Não foi o caso da implementação que é objeto desta dissertação.

No Projeto Cyclops já existe muito software para suporte ao padrão DICOM desenvolvido, mas um servidor era algo que ainda estava faltando. Muito do que é necessário para a implementação deste servidor pode ser (o foi e o será) reutilizado do que já existe no projeto. Muitas pessoas, no Brasil e na Alemanha, trabalharam para desenvolver estes softwares para DICOM dentro do Cyclops. Durante o desenvolvimento desta dissertação pode-se verificar que algumas partes do que já foi implementado estão confusas, mas outras estão muito bem implementadas. Este trabalho certamente será o responsável pela reformulação de muito deste software já desenvolvido no projeto. Certamente ter-se-á em mãos brasileiras (e germânicas) um conjunto de softwares para suporte ao padrão DICOM, robusto e em conformidade, cujas tecnologias estarão disponíveis aos respectivos interessados.

O real objetivo por trás deste protótipo é o acréscimo de mais um recurso aos aplicativos do Projeto Cyclops, visando a maior aceitação dos mesmos por parte dos seus usuários nos ambientes clínico-hospitalares. Com este trabalho, o passo inicial para tal foi dado.

Acredita-se que o objetivo deste trabalho tenha sido alcançado, pois as técnicas necessárias para a implementação do protótipo proposto como objetivo foram pesquisadas, estudadas e finalmente postas em prática. Porém, o que se tem neste momento é apenas um protótipo, e muito trabalho ainda falta para que se tenha um aplicativo robusto e confiável que possa ser utilizado dentro de ambientes clínico-hospitalares, sem oferecer riscos a todos os usuários de serviços destes ambientes.

Anexo 1 – Código em Smalltalk para iniciar o servidor



Anexo 2 - Saída do comando “storescu”

```
cyclone:~$ /opt/dicom/bin/storescu +v cyclone 4006 ct*.dcm
Request Parameters:
Our Implementation Class UID:      1.2.276.0.7230010.3.0.3.4.2
Our Implementation Version Name:  OFFIS_DCMTK_342
Their Implementation Class UID:
Their Implementation Version Name:
Application Context Name:         1.2.840.10008.3.1.1.1
Calling Application Name:         STORESCU
Called Application Name:          ANY-SCP
Responding Application Name:      resp AP Title
Our Max PDU Receive Size: 16384
Their Max PDU Receive Size: 0
Presentation Contexts:
  Context ID:          1 (Proposed)
    Abstract Syntax:    =AmbulatoryECGWaveformStorage
    Proposed SCP/SCU Role: Default
    Accepted SCP/SCU Role: Default
    Proposed Transfer Syntax(es):
      =LittleEndianExplicit

(...)

Context ID:          13 (Proposed)
  Abstract Syntax:    =CTImageStorage
  Proposed SCP/SCU Role: Default
  Accepted SCP/SCU Role: Default
  Proposed Transfer Syntax(es):
    =LittleEndianExplicit
Context ID:          15 (Proposed)
  Abstract Syntax:    =CTImageStorage
  Proposed SCP/SCU Role: Default
  Accepted SCP/SCU Role: Default
  Proposed Transfer Syntax(es):
    =BigEndianExplicit
    =LittleEndianImplicit

(...)

Context ID:          221 (Proposed)
  Abstract Syntax:    =XRayFluoroscopyImageStorage
  Proposed SCP/SCU Role: Default
  Accepted SCP/SCU Role: Default
  Proposed Transfer Syntax(es):
    =LittleEndianExplicit
Context ID:          223 (Proposed)
  Abstract Syntax:    =XRayFluoroscopyImageStorage
  Proposed SCP/SCU Role: Default
  Accepted SCP/SCU Role: Default
  Proposed Transfer Syntax(es):
    =BigEndianExplicit
    =LittleEndianImplicit
Requested Extended Negotiation: none
Accepted Extended Negotiation: none
```

Requesting Association

Association Parameters Negotiated:

Our Implementation Class UID: 1.2.276.0.7230010.3.0.3.4.2

Our Implementation Version Name: OFFIS_DCMTK_342

Their Implementation Class UID: 6.666.6.66.6666.666.66.6

Their Implementation Version Name: Cyclops DCM

Application Context Name: 1.2.840.10008.3.1.1.1

Calling Application Name: STORESCU

Called Application Name: ANY-SCP

Responding Application Name: ANY-SCP

Our Max PDU Receive Size: 16384

Their Max PDU Receive Size: 16384

Presentation Contexts:

Context ID: 1 (Abstract Syntax Not Supported)

Abstract Syntax: =AmbulatoryECGWaveformStorage

Proposed SCP/SCU Role: Default

Accepted SCP/SCU Role: Default

(...)

Context ID: 13 (Transfer Syntaxes Not Supported)

Abstract Syntax: =CTImageStorage

Proposed SCP/SCU Role: Default

Accepted SCP/SCU Role: Default

Context ID: 15 (Accepted)

Abstract Syntax: =CTImageStorage

Proposed SCP/SCU Role: Default

Accepted SCP/SCU Role: Default

Accepted Transfer Syntax: =LittleEndianImplicit

(...)

Context ID: 221 (Abstract Syntax Not Supported)

Abstract Syntax: =XRayFluoroscopyImageStorage

Proposed SCP/SCU Role: Default

Accepted SCP/SCU Role: Default

Context ID: 223 (Abstract Syntax Not Supported)

Abstract Syntax: =XRayFluoroscopyImageStorage

Proposed SCP/SCU Role: Default

Accepted SCP/SCU Role: Default

Requested Extended Negotiation: none

Accepted Extended Negotiation: none

Association Accepted (Max Send PDV: 16372)

Sending file: ct1.dcm

Transfer: LittleEndianImplicit -> LittleEndianImplicit

Store SCU RQ: MsgID 1, (CT)

XMIT:.....

C-Store RSP: MsgID: 1 [Status=Success]

AffectedSOPClassUID: =CTImageStorage

AffectedSOPInstanceUID: 1.2.840.113619.2.22.287.1.19990420.262632

Data Set: Not Present

Sending file: ct2.dcm

Transfer: LittleEndianImplicit -> LittleEndianImplicit

Store SCU RQ: MsgID 2, (CT)

XMIT:.....

C-Store RSP: MsgID: 2 [Status=Success]

AffectedSOPClassUID: =CTImageStorage

```
AffectedSOPInstanceUID: 1.2.840.113619.2.22.287.1.19990420.262632
Data Set: Not Present
-----
Sending file: ct3.dcm
Transfer: LittleEndianImplicit -> LittleEndianImplicit
Store SCU RQ: MsgID 3, (CT)
XMIT:.....
C-Store RSP: MsgID: 3 [Status=Success]
  AffectedSOPClassUID: =CTImageStorage
  AffectedSOPInstanceUID: 1.2.840.113619.2.22.287.1.19990420.262638
  Data Set: Not Present
Releasing Association
cyclone::~$
```

Anexo 3 – Saída do comando “send_image”

```
cyclone:~$ /opt/ctn-2.11.3/bin/send_image cyclone 4006 ct*.dcm
Association accepted, parameters:
APP CTX NAME:1.2.840.10008.3.1.1.1
                Application Context Name, NEMA
AP TITLE:      DICOM_TEST
AP TITLE:      DICOM_STORAGE
AP TITLE:      DICOM_STORAGE
MAX PDU:       16384
Peer MAX PDU:  16384
PRES ADDR:     cyclone
PRES ADDR:     cyclone:4006
REQ IMP UID:   1.2.840.113654.2.3.1995.2.11.3
                Implementation Class UID, MIR
REQ VERSION:   MIRCTN12MAY2000
ACC IMP UID:   6.666.6.66.6666.666.66.6
                Unknown UID
ACC VERSION:   Cyclops DCM
Requested Presentation Ctx
  Context ID:   1
  Abstract Syntax: 1.2.840.10008.5.1.4.1.1.2
                CT Image Storage, NEMA
  Result field: 0
  Proposed SCU/SCP Role: SCU
  Accepted SCU/SCP Role: Default
  Proposed Xfer Syntax(es)
                1.2.840.10008.1.2
                Implicit Little-Endian Transfer Syntax, NEMA
  Accepted Xfer Syntax:
                No UID
Accepted Presentation Ctx
  Context ID:   1
  Abstract Syntax: 1.2.840.10008.5.1.4.1.1.2
                CT Image Storage, NEMA
  Result field: 0
  Proposed SCU/SCP Role: SCU
  Accepted SCU/SCP Role: Default
  Proposed Xfer Syntax(es)
  Accepted Xfer Syntax: 1.2.840.10008.1.2
                Implicit Little-Endian Transfer Syntax, NEMA
Initial call to sendCallback
  0 bytes transmitted of 526302 (context string)
  16364 bytes transmitted of 526302 (context string)
  32728 bytes transmitted of 526302 (context string)
  49092 bytes transmitted of 526302 (context string)
  65456 bytes transmitted of 526302 (context string)
  81820 bytes transmitted of 526302 (context string)
  98184 bytes transmitted of 526302 (context string)
  114548 bytes transmitted of 526302 (context string)
  130912 bytes transmitted of 526302 (context string)
  147276 bytes transmitted of 526302 (context string)
  163640 bytes transmitted of 526302 (context string)
  180004 bytes transmitted of 526302 (context string)
  196368 bytes transmitted of 526302 (context string)
```

```

212732 bytes transmitted of 526302 (context string)
229096 bytes transmitted of 526302 (context string)
245460 bytes transmitted of 526302 (context string)
261824 bytes transmitted of 526302 (context string)
278188 bytes transmitted of 526302 (context string)
294552 bytes transmitted of 526302 (context string)
310916 bytes transmitted of 526302 (context string)
327280 bytes transmitted of 526302 (context string)
343644 bytes transmitted of 526302 (context string)
360008 bytes transmitted of 526302 (context string)
376372 bytes transmitted of 526302 (context string)
392736 bytes transmitted of 526302 (context string)
409100 bytes transmitted of 526302 (context string)
425464 bytes transmitted of 526302 (context string)
441828 bytes transmitted of 526302 (context string)
458192 bytes transmitted of 526302 (context string)
474556 bytes transmitted of 526302 (context string)
490920 bytes transmitted of 526302 (context string)
507284 bytes transmitted of 526302 (context string)
523648 bytes transmitted of 526302 (context string)
526194 bytes transmitted of 526302 (context string)
526302 bytes transmitted of 526302 (context string)
Store Response
Message ID Resp:1
Data Set Type: 0101
Status: 0000 Status Information:-
    Successful operation
Class UID: 1.2.840.10008.5.1.4.1.1.2
Instance UID: 1.2.840.113619.2.22.287.1.19990420.262632
Store Response
Message ID Resp:1
Data Set Type: 0101
Status: 0000 Status Information:-
    Successful operation
Class UID: 1.2.840.10008.5.1.4.1.1.2
Instance UID: 1.2.840.113619.2.22.287.1.19990420.262632
Initial call to sendCallback
    0 bytes transmitted of 526302 (context string)
    16364 bytes transmitted of 526302 (context string)
    32728 bytes transmitted of 526302 (context string)
    49092 bytes transmitted of 526302 (context string)
    65456 bytes transmitted of 526302 (context string)
    81820 bytes transmitted of 526302 (context string)
    98184 bytes transmitted of 526302 (context string)
    114548 bytes transmitted of 526302 (context string)
    130912 bytes transmitted of 526302 (context string)
    147276 bytes transmitted of 526302 (context string)
    163640 bytes transmitted of 526302 (context string)
    180004 bytes transmitted of 526302 (context string)
    196368 bytes transmitted of 526302 (context string)
    212732 bytes transmitted of 526302 (context string)
    229096 bytes transmitted of 526302 (context string)
    245460 bytes transmitted of 526302 (context string)
    261824 bytes transmitted of 526302 (context string)
    278188 bytes transmitted of 526302 (context string)
    294552 bytes transmitted of 526302 (context string)
    310916 bytes transmitted of 526302 (context string)
    327280 bytes transmitted of 526302 (context string)
    343644 bytes transmitted of 526302 (context string)

```

```

360008 bytes transmitted of 526302 (context string)
376372 bytes transmitted of 526302 (context string)
392736 bytes transmitted of 526302 (context string)
409100 bytes transmitted of 526302 (context string)
425464 bytes transmitted of 526302 (context string)
441828 bytes transmitted of 526302 (context string)
458192 bytes transmitted of 526302 (context string)
474556 bytes transmitted of 526302 (context string)
490920 bytes transmitted of 526302 (context string)
507284 bytes transmitted of 526302 (context string)
523648 bytes transmitted of 526302 (context string)
526194 bytes transmitted of 526302 (context string)
526302 bytes transmitted of 526302 (context string)
Store Response
Message ID Resp:2
Data Set Type: 0101
Status: 0000 Status Information:-
    Successful operation
Class UID: 1.2.840.10008.5.1.4.1.1.2
Instance UID: 1.2.840.113619.2.22.287.1.19990420.262632
Store Response
Message ID Resp:2
Data Set Type: 0101
Status: 0000 Status Information:-
    Successful operation
Class UID: 1.2.840.10008.5.1.4.1.1.2
Instance UID: 1.2.840.113619.2.22.287.1.19990420.262632
Initial call to sendCallback
    0 bytes transmitted of 526306 (context string)
    16364 bytes transmitted of 526306 (context string)
    32728 bytes transmitted of 526306 (context string)
    49092 bytes transmitted of 526306 (context string)
    65456 bytes transmitted of 526306 (context string)
    81820 bytes transmitted of 526306 (context string)
    98184 bytes transmitted of 526306 (context string)
    114548 bytes transmitted of 526306 (context string)
    130912 bytes transmitted of 526306 (context string)
    147276 bytes transmitted of 526306 (context string)
    163640 bytes transmitted of 526306 (context string)
    180004 bytes transmitted of 526306 (context string)
    196368 bytes transmitted of 526306 (context string)
    212732 bytes transmitted of 526306 (context string)
    229096 bytes transmitted of 526306 (context string)
    245460 bytes transmitted of 526306 (context string)
    261824 bytes transmitted of 526306 (context string)
    278188 bytes transmitted of 526306 (context string)
    294552 bytes transmitted of 526306 (context string)
    310916 bytes transmitted of 526306 (context string)
    327280 bytes transmitted of 526306 (context string)
    343644 bytes transmitted of 526306 (context string)
    360008 bytes transmitted of 526306 (context string)
    376372 bytes transmitted of 526306 (context string)
    392736 bytes transmitted of 526306 (context string)
    409100 bytes transmitted of 526306 (context string)
    425464 bytes transmitted of 526306 (context string)
    441828 bytes transmitted of 526306 (context string)
    458192 bytes transmitted of 526306 (context string)
    474556 bytes transmitted of 526306 (context string)
    490920 bytes transmitted of 526306 (context string)

```

```
507284 bytes transmitted of 526306 (context string)
523648 bytes transmitted of 526306 (context string)
526198 bytes transmitted of 526306 (context string)
526306 bytes transmitted of 526306 (context string)
Store Response
Message ID Resp:3
Data Set Type: 0101
Status: 0000 Status Information:-
    Successful operation
Class UID: 1.2.840.10008.5.1.4.1.1.2
Instance UID: 1.2.840.113619.2.22.287.1.19990420.262638
Store Response
Message ID Resp:3
Data Set Type: 0101
Status: 0000 Status Information:-
    Successful operation
Class UID: 1.2.840.10008.5.1.4.1.1.2
Instance UID: 1.2.840.113619.2.22.287.1.19990420.262638
cyclone:~$
```

Anexo 4 – Saída do comando “echoscu”

```
cyclone:~$ /opt/dicom/bin/echoscu -v cyclone 4006  
Requesting Association  
Association Accepted (Max Send PDV: 16372)  
Echo [1], Complete [Status: Success]  
Releasing Association  
cyclone:~$
```


Anexo 5 – Saída do comando “dicom_echo”

```
cyclone:~$ /opt/ctn-2.11.3/bin/dicom_echo -v cyclone 4006
DUL_InitializeNetwork, Type: NETWORK TCP, Mode: AE REQUESTOR
DUL_Request Association
DUL FSM Table: State: 1 Event: 0
DUL Event: A-ASSOCIATE request (local user)
DUL Action: AE 1 Transport Connect
DUL FSM Table: State: 4 Event: 1
DUL Event: Transport conn confirmation (local)
DUL Action: AE 2 Send Associate RQ PDU
Constructing Associate RQ PDU
Read PDU HEAD TCP: 02 00 00 00 00 b1
Read PDU HEAD TCP: type: 2, length: 177 (b1)
DUL FSM Table: State: 5 Event: 2
DUL Event: A-ASSOCIATE-AC PDU (on transport)
DUL Action: AE 3 Associate Confirmation Accept
PDU Type: Associate Accept PDU Length: 183
  02 00 00 00 00 b1 00 01 00 00 44 49 43 4f 4d 5f
  53 54 4f 52 41 47 45 20 20 20 44 49 43 4f 4d 5f
  45 43 48 4f 20 20 20 20 20 20 00 00 00 00 00 00
  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  00 00 00 00 00 00 00 00 00 00 10 00 00 15 31 2e
  32 2e 38 34 30 2e 31 30 30 30 38 2e 33 2e 31 2e
  31 2e 31 21 00 00 19 01 00 00 00 40 00 00 11 31
  2e 32 2e 38 34 30 2e 31 30 30 30 38 2e 31 2e 32
  50 00 00 33 51 00 00 04 00 00 40 00 52 00 00 18
  36 2e 36 36 36 2e 36 2e 36 36 2e 36 36 36 36 2e
  36 36 36 2e 36 36 2e 36 55 00 00 0b 43 79 63 6c
  6f 70 73 20 44 43 4d
Parsing an A-ASSOCIATE PDU
PDU type: 2 (A-ASSOCIATE AC), PDU Length: 177
DICOM Protocol: 1
Called AP Title: DICOM_STORAGE
Calling AP Title: DICOM_ECHO
Parsing remaining 109 bytes of A-ASSOCIATE PDU
Next item type: 10
Subitem parse: Type 10, Length 21, Content: 1.2.840.10008.3.1.1.1
Successfully parsed Application Context
Parsing remaining 84 bytes of A-ASSOCIATE PDU
Next item type: 21
Parsing Presentation Context: (21), Length: 25
Presentation Context ID: 1
Parsing remaining 21 bytes of Presentation Ctx
Next item type: 40
Subitem parse: Type 40, Length 17, Content: 1.2.840.10008.1.2
Successfully parsed Transfer Syntax
Successfully parsed Presentation Context
Parsing remaining 55 bytes of A-ASSOCIATE PDU
Next item type: 50
Parsing user info field (50), Length: 51
Parsing remaining 51 bytes of User Information
Next item type: 51
Maximum PDU Length: 16384
Successfully parsed Maximum PDU Length
```

```

Parsing remaining 43 bytes of User Information
Next item type: 52
Subitem parse: Type 52, Length 24, Content: 6.666.6.66.6666.666.66.6
Parsing remaining 15 bytes of User Information
Next item type: 55
Subitem parse: Type 55, Length 11, Content: Cyclops DCM
Successfully parsed User Information
Association accepted, parameters:
APP CTX NAME:1.2.840.10008.3.1.1.1
      Application Context Name, NEMA
AP TITLE:      DICOM_ECHO
AP TITLE:      DICOM_STORAGE
AP TITLE:      DICOM_STORAGE
MAX PDU:       16384
Peer MAX PDU:  16384
PRES ADDR:     cyclone
PRES ADDR:     cyclone:4006
REQ IMP UID:   1.2.840.113654.2.3.1995.2.11.3
      Implementation Class UID, MIR
REQ VERSION:   MIRCTN12MAY2000
ACC IMP UID:   6.666.6.66.6666.666.66.6
      Unknown UID
ACC VERSION:   Cyclops DCM
Requested Presentation Ctx
  Context ID:      1
  Abstract Syntax:  1.2.840.10008.1.1
      Verification SOP Class, NEMA
  Result field:    0
  Proposed SCU/SCP Role: Default
  Accepted SCU/SCP Role: Default
  Proposed Xfer Syntax(es)
      1.2.840.10008.1.2
      Implicit Little-Endian Transfer Syntax, NEMA
  Accepted Xfer Syntax:
      No UID
Accepted Presentation Ctx
  Context ID:      1
  Abstract Syntax:  1.2.840.10008.1.1
      Verification SOP Class, NEMA
  Result field:    0
  Proposed SCU/SCP Role: Default
  Accepted SCU/SCP Role: Default
  Proposed Xfer Syntax(es)
  Accepted Xfer Syntax: 1.2.840.10008.1.2
      Implicit Little-Endian Transfer Syntax, NEMA
DUL FSM Table: State: 6 Event: 8
DUL Event: P-DATA request primitive
DUL Action: DT 1 Send P DATA PDU
Beginning to read a COMMAND PDV
Read PDU HEAD TCP: 04 00 00 00 00 54
Read PDU HEAD TCP: type: 4, length: 84 (54)
DUL FSM Table: State: 6 Event: 9
DUL Event: P-DATA-TF PDU (on transport)
DUL Action: DT 2 Indicate P DATA PDU Received
  00 00 00 50 01 03 00 00 00 00 04 00 00 00 42 00
  00 00 00 00 02 00 12 00 00 00 31 2e 32 2e 38 34
  30 2e 31 30 30 30 38 2e 31 2e 31 00 00 00 00 01
  02 00 00 00 30 80 00 00 20 01 02 00 00 00 01 00
  00 00 00 08 02 00 00 00 01 01 00 00 00 09 02 00

```

```

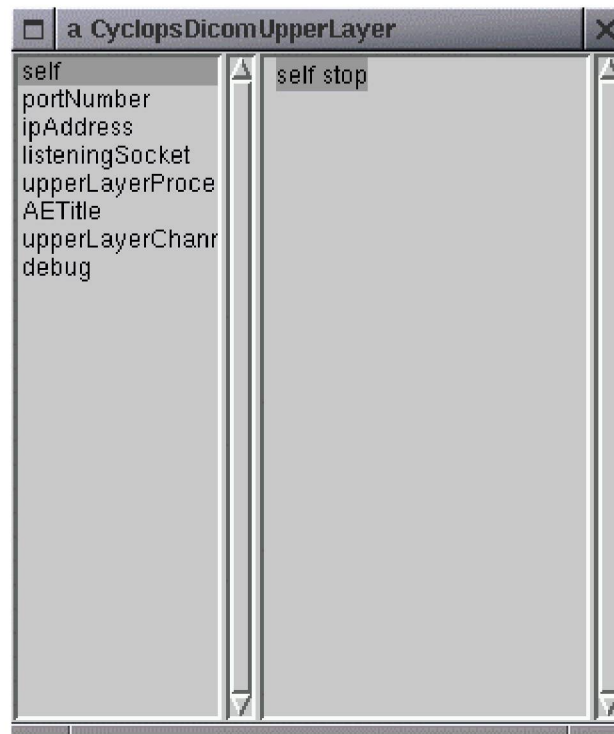
00 00 00 00

DCM Dump Elements
Object type: COMMAND
Object size: 78
Group: 0000, Length:      66
0000 0000      4 //          CMD Group Length//      42 66
0000 0002      18 //      CMD Affected SOP Class
UID//1.2.840.10008.1.1
0000 0100      2 //          CMD Command Field//8030 32816
0000 0120      2 //      CMD Message ID Responded to//  1 1
0000 0800      2 //          CMD Data Set Type// 101 257
0000 0900      2 //          CMD Status//  0 0
DCM Dump Elements Complete

Command received: 8030
Echo context: Context
Verification Response
  Message ID Responded to:  1
  Verification Status:      0000
Echo Response
Message ID Responded To: 1
Data Set Type:      0101
Status:      0000  Status Information:-
  Successful operation
Class UID:      1.2.840.10008.1.1
DUL  FSM Table: State:  6 Event: 10
DUL  Event:  A-RELEASE request primitive
DUL  Action: AR 1 Send Release RQ
Read PDU HEAD TCP:  06 00 00 00 00 04
Read PDU HEAD TCP: type: 6, length: 4 (4)
DUL  FSM Table: State:  7 Event: 12
DUL  Event:  A-RELEASE-RP PDU (on transport)
DUL  Action: AR 3 Confirm Release
DUL DropNetwork
cyclone:~$

```

Anexo 6 – Código em Smalltalk para parar o servidor



7 - Glossário

ACR	American College of Radiology
ACSE	Association Control Service Element
AE	Application Entity
ARTIM	Association Request/Reject/Release Timer
CT	Computed Tomography
CTN	Central Test Node
DICOM	Digital Image Communications on Medicine
DICOM-UL	DICOM Upper Layer
DIMSE	DICOM Message Service Element
DIMSE-C	Composite DIMSE
DIMSE-M	Media DIMSE
DIMSE-N	Normalized DIMSE
FDDI	Fiber Distributed Data Interface
HIS	Hospital Information Systems
IE	Information Entity
IOD	Information Object Definition
IOM	Information Object Module
IP	Internet Protocol
ISDN	Integrated Services Digital Network
LLC	Logical Link Layer
MR	Magnetic Ressonance
NEMA	National Electrical Manufacturers Association
NM	Nuclear Medicine

OSI	Open Systems Interface
PACS	Picture Archiving and Communications Systems
PDU	Protocol Data Unit
PSAP	OSI Presentation Service Access Point
SAP	Service Access Point
SCP	Service Class Provider
SCU	Service Class User
SOP	Service-Object Pair
SQL	Simple Query Language
TCP/IP	Transfer Control Protocol/Internet Protocol
UID	Unique Identifier
US	Ultra-Sound
VM	Value Multiplicity
VR	Value Representation

8 - Referências Bibliográficas

BECK, Kent. **SMALLTALK: BEST PRACTICE PATTERNS**. New Jersey: Prentice Hall, 1997. 224 p.

BRAMORSKI, Mônica Maria. Extensão ao Protocolo DICOM para Suportar Realidade Virtual: Modelagem do *Virtual Reality Information Object Definition* (VR IOD). 1999. 74 f. Relatório (Bacharel em Ciência da Computação) – Curso de Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis.

CARVALHO, Tereza C.M.B. **Arquiteturas de Redes de Computadores OSI e TCP/IP**. Makron Books, Rio de Janeiro, 1994.

CLUNIE, David A. **Medical Image Format FAQ**. Questões freqüentemente perguntadas sobre Formatos de Imagens Médicas, com enfoque para o padrão DICOM. Disponível em: <<http://www.dclunie.com/medical-image-faq/html/>>. Acesso em: 10 de novembro de 2000.

EFILM MEDICAL INC. **eFilm Workstation 1.5.3 Software** . Disponível em: <<http://www.efilm.ca/Downloads/Download.asp?product=1&version=1.5.3SP1>>. Acesso em: 05 de fevereiro de 2001.

HOPKINS, Trevor; HORAN, Bernard. **SMALLTALK: An Introduction to Application Development Using Visualworks**. Oxford: Prentice Hall, 1995.

MALLINCKRODT INSTITUTE OF RADIOLOGY (MIR). **DICOM Central Test Node 3.0 Software**. Disponível em: <<ftp://ftp.erl.wustl.edu/pub/dicom/software/ctn/ctn-3.0.0.tar.Z>>. Acesso em: 20 de fevereiro de 2001.

McCLAIN, Gary R. **Handbook of Network & Connectivity**. Cambridge: Academic Press, 1994. 415 p.

MERGE TECHNOLOGIES INC. **The Official DICOM Resources Page**: Presented by the Committee for the Advancement of DICOM. This page presents an evolving set of resources available to everyone interested in the DICOM standard or its implementation. Disponível em: <<http://www.merge.com/dicom/resources/>>. Acesso em: 25 de novembro de 2000.

MURRAY, James D.; van RYPER, Willian. **Encyclopedia of Graphics File Formats**. 2.ed. Cambridge: O'Reilly & Associates, Inc., 1996. 1116 p.

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. **Digital Imaging and Communications in Medicine (DICOM) - Part 1: Introduction and Overview**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_01dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 2: Conformance**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_02dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_03dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 4: Service Class Specifications**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_04dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 5: Data Structures and Encoding**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_05dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 6: Data Dictionary**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_06dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 7: Message Exchange**. Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_07dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 8: Network Communication Support for Message Exchange**. Virginia, 2000.

Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_08dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 10: Media Storage and File Format for Media Interchange.** Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_10dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 11: Media Storage Application Profiles.** Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_11dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 12: Media Formats and Physical Media for Media Interchange.** Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_12dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 13: Print Management Point-to-Point Communication Support.** Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_13dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 14: Grayscale Standard Display Function.** Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_14dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **Digital Imaging and Communications in Medicine (DICOM) - Part 15: Security Profiles.** Virginia, 2000. Disponível em: <ftp://medical.nema.org/medical/dicom/2000/draft/00_15dr.pdf>. Acesso em: 10 de novembro de 2000.

_____. **NEMA's OFFICIAL DICOM WEB PAGE.** Disponível em: <<http://medical.nema.org/dicom.html>>. Acesso em: 20 de novembro de 2000.

OFFIS. DICOM: Medical Image Communication Standardisation. Disponível em: <http://www.offis.uni-oldenburg.de/projekte/dicom/project_dicom.htm>. Acesso em: 10 de dezembro de 2000.

RADIOLOGICAL SOCIETY OF NORTH AMERICA, INC. **DICOM: The Value and Importance of an Imaging Standard**. Disponível na Internet via <<http://www.rsna.org/REG/practiceres/dicom/index.html>>. Acesso em: 23 de janeiro de 2001.

RENET, Bas. **DICOM Cook Book for Implementations in Modalities. Chapters 1 and 2**. Netherlands: Philips Medical Systems, January 1997. Disponível em: <ftp://ftp-wjq.philips.com/medical/interoperability/out/DICOM_Information/CookBook.pdf>. Acesso em: 10 de agosto de 2000.

SAMPAIO, Silvio Costa. **Modelagem e Implementação Orientada a Objetos de um Cliente de Rede para Banco de Dados de Imagens Médicas Digitais Utilizando o Padrão DICOM 3.0**. 1999. 90 f. Dissertação (Mestrado em Ciência da Computação) – Curso de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis.

STEVENS, W.Richard. **UNIX Network Programming Volume 1 Networking APIs: Sockets and XTI**. 2.ed. New Jersey: Prentice Hall, 1998. 1009 p.

UNIVERSIDADE DE PISA. **DICOM Group Homepage**. Disponível em: <<http://dicom.rad.unipi.it>>. Acesso em: 01 de agosto de 2000.

UNIVERSITY HOSPITAL OF GENEVA. **The Osiris Imaging Software**. Disponível em: <<http://www.expasy.ch/www/UIN/html1/projects/osiris/osiris.html>>. Acesso em: 01 de fevereiro de 2001.

_____. **Osiris Help**. Documento descrevendo como utilizar o software OSIRIS. Disponível em: <<http://www.expasy.ch/www/UIN/html1/projects/osiris/osirismanual.html>>. Acesso em: 01 de fevereiro de 2001.

_____. **PAPYRUS 3.0: A DICOM compatible file format**. Disponível em: <<http://www.expasy.ch/UIN/html1/projects/papyrus/papyrus.html>>. Acesso em: 01 de fevereiro de 2001.

UNIVERSITY OF UTRECHT. Department of Computer Sciences. **Medical Image Format FAQ from Newsgroups**. Disponível em: <<http://www.cs.ruu.nl/wais/html/nabng/comp.protocols.dicom.html>>. Acesso em 10 de novembro de 2000.